PH.D.

Finding Structure in Language

Steven Finch

University of Edinburgh
1993

Declaration

I declare that this thesis has been composed by myself and that the research reported therein has been conducted by myself unless otherwise indicated.

Steven Finch

10th August 1993

Abstract

Since the Chomskian revolution, it has become apparent that natural language is richly structured, being naturally represented hierarchically, and requiring complex context sensitive rules to define regularities over these representations. It is widely assumed that the richness of the posited structure has strong nativist implications for mechanisms which might learn natural language, since it seemed unlikely that such structures could be derived directly from the observation of linguistic data (Chomsky 1965).

This thesis investigates the hypothesis that simple statistics of a large, noisy, unlabelled corpus of natural language can be exploited to discover some of the structure which exists in natural language automatically. The strategy is to initially assume no knowledge of the structures present in natural language, save that they might be found by analysing statistical regularities which pertain between a word and the words which typically surround it in the corpus.

To achieve this, various statistical methods are applied to define similarity between statistical distributions, and to infer a structure for a domain given knowledge of the similarities which pertain within it. Using these tools, it is shown that it is possible to form a hierarchical classification of many domains, including words in natural language. When this is done, it is shown that all the major syntactic categories can be obtained, and the classification is both relatively complete, and very much in accord with a standard linguistic conception of how words are classified in natural language.

Once this has been done, the categorisation derived is used as the basis of a similar classification of short sequences of words. If these are analysed in a similar way, then several syntactic categories can be derived. These include simple noun phrases, various tensed forms of verbs, and simple prepositional phrases. Once this has been done, the same technique can be applied one level higher, and at this level simple sentences and verb phrases, as well as more complicated noun phrases and prepositional phrases, are shown to be derivable.

Acknowledgements

In preparing this Ph.D., I have had help and useful suggestions from many people. I would like to thank some of them here although I am sure the list will be incomplete.

Firstly, I thank my parents for their moral and substantial support throughout my life.

I would like to thank those responsible for the stimulating interdisciplinary research environment here at the Centre for Cognitive Science at Edinburgh, and those who provide the computing infrastructure which made this work possible.

Thanks to my supervisors: Terry Myers, David Willshaw and Nick Chater. David has been a great encouragement during this research, and has contributed many substantive suggestions which have been incorporated into this work, including the suggestion that hierarchical cluster analysis be used to explicate structure. Nick has not only made many invaluable and substantive suggestions, but has collaborated with me in writing the papers which have been published as a result of this research. The ideas he contributed have had a large influence on the shape of the research contained in this thesis. I thank him also for the considerable amount of time and energy he has devoted to carefully reading and commenting on drafts of this thesis. I thank Terry for originating my interest in unsupervised learning.

Thanks to Sophie Cormack for her friendship, support and chocolate cake, as well as many stimulating conversations on all subjects. Thanks too, in no particular order, to Peter Dayan, Guy Barry, Mark Ellison, Martin Pickering, Jerry Seligman, Mike Oaksford, Mike Malloch and Nick Chater again (in his previous guise as a PhD student) for the many conversations which helped shape my views on the relationship between statistics, connectionism, learning, language and philosophy.

Finally, I thank Cally for her invaluable support and help during these last six months, and for showing me some other sides to life.

Contents

Chapter 1

Introduction: Motivation and Outline

This thesis is the result of three years of research undertaken here at the University of Edinburgh, and this chapter is the last to be written. It describes some of the interests I had in cognitive science which motivated my research, and outlines the chapters which follow. Finally, a brief sketch of related and background academic work is presented.

1.1 Learning Sophisticated Theories

How can an agent, such as a human being, a cat, or a computer, come to acquire a sophisticated, richly structured theory of the various processes which operate in the world? This general question motivates this thesis, and it is interesting because it addresses many problems in cognitive science. It is relevant to cognitive psychology (how do *we* acquire such theories), linguistics (what is the nature of theories of language), computational linguistics (how can linguistic competence be acquired by a computer), and philosophy (how can a theory which makes generalisations be induced from a finite sample of data?). Thus it is pertinent to all the main branches of cognitive science.

From introspection, of which one always needs to be wary, it appears that we have highly sophisticated representations and theories of the various processes which make up the

world in which we find ourselves situated. When we see a photograph in a newspaper, we automatically interpret it as a representation of a real world situation, splitting the various forms in the picture into objects, people, faces, streets, and so on. And yet none of this is inherent in the photograph itself, which is just a collection of various sized and coloured blobs on newsprint. When we hear a conversation over the radio, we interpret the sound as speech, splitting it into words, and identifying which words were said by who, and yet none of this is inherent in the representation of that sound wave transmitted via a radio signal, which is just a representation of small variations in air pressure picked up by a microphone. When we read a text we interpret what we read into linguistic units such as nouns, verbs, sentences, and the like, which we further interpret and understand as referring to various events and facts about the outside world. And yet none of this is inherent in the text which is written, since this is just a sequence of alpha-numeric characters. Yet all of these simple representations contain enough information for us to be able to derive a far more sophisticated, structured representation of the same information, which interfaces with the various sophisticated theories we have about the world and allows us to understand and draw conclusions which are useful to us. We *know* they contain enough information because we *can* process the information and perform useful inferences using it. Moreover, such simple representations of information are generally and cheaply available to researchers to perform research with.

In order to start to find an answer to how we acquire, or can produce a machine which acquires, such sophisticated representations and theories, one needs to ask what information or biases such agents have at their disposal which facilitates this process. One promptly comes to the conclusion that there are two sorts of information — information from the world obtained via systems which are sensitive to real world processes external to the agent, such as sight, hearing, taste, touch, and so on; and innate biases (in animals, through genetic information, in computers through programming) which tell the agent how to interpret this information, and how to behave on the basis of such interpretation.

The same dichotomy of available information is true for the derivation of scientific theories. We have available some empirical observations of real-world processes (which

correspond to sensory information), and possess certain beliefs about the structure of these processes (which correspond, roughly, to innate information). Science can also be useful in answering the question of what *makes* a good sophisticated representation, or a good sophisticated theory. An example of a sophisticated theory in science is the hypothesis of the particulate nature of matter. Atoms do not make themselves known to investigators directly — their existence is demonstrated by the utility of various theories which assume their existence in making valid generalisations and predictions about physical processes the scientist can observe and perform experiments on. Another example is the heliocentric theory of the solar system after Galileo. Long before man had more direct knowledge of the structure of the solar system, the heliocentric theory correctly predicted (and hence explained) facts about the seasonal distribution of sunlight on the earth at various times of the year, the fact that some planets sometimes appear to change direction and move in a 'retrograde' manner, as well as giving good explanations for the prima facie contrary evidence that the sun appears to circle the earth. Moreover, the heliocentric theory could easily be interpreted in a general theory of gravitation proposed (later) by Newton, which made highly accurate predictions about the motion of the planets. Indeed, loose apples notwithstanding, it would seem hard to imagine how Newton could have come up with and tested his theory of gravitation without the heliocentric theory of the solar system as a prior context. In the example of the particulate nature of matter, atoms are hidden and their existence must be deduced. In the example of the heliocentric theory of the solar system, assuming such a representation of the solar system allowed the direct application (and possibly the derivation) of a general theory of matter, facilitating more accurate predictions.

Often, an agent has access only to a small potion of the data from a physical process it wishes to find a theory of. For instance, our knowledge of the outside world is mediated by our senses, prime among which are sight and sound. From the evidence of these senses, we are able to infer a sophisticated object-based representation of the physical world, which is just the sort of representation structure needed if we are to manipulate objects in the world, avoid dangerous situations, find food, and perform all the other tasks helpful to our survival. There are two diametrically opposed views as to how we might acquire such a theory.

The first is a radically nativist position — such sophisticated theories are not learned, rather they are innate, and the acquisition of such theories is largely a process of maturation, rather than a process of data-dependent learning, this process having been 'pre-programmed' by natural selection. This position is that held by Fodor in *The Language of Thought* (1975), and an extreme version of Chomsky's (1965) innateness hypothesis, which was more extremely stated by Chomsky himself in *Reflections* (1975). In the analogy of the computational theory of mind, it roughly states that even if the program to cognize is not present at first,[1] there is an automatic 'program modification system', which automatically modifies the existing algorithm for cognition without reference to external data, and which will make the correct sophisticated theory available to the agent at some future time with little contingency on sensory information. Also, the ability to find a sophisticated representation must be innate, because no process of learning can change the representational power of the system. Clearly, even in this view, the role of data-dependent learning cannot be eliminated — children are, after all, capable of learning any one of a multitude of languages, but do not learn languages to which they are not exposed — but the role of such data-dependent learning within processes of acquisition is minimised.

The second is a radically empiricist position. In this view, knowledge of everything to do with the world is learned[2] from some computational analysis of the data observed by the senses. This is roughly the position of John Locke (1690) who believed that the infant was as a blank sheet of paper upon which the experience of the senses would write its teachings. In this view, the process of finding sophisticated representations and theories is one of discovery from the evidence presented by the senses, rather than a pre-programmed description of these representations and theories. Again, in the light of what we now know about human learning, and the complexity of even specific instances of induction, no one would today say that entirely unconstrained empirical induction was a viable proposal as it stands, but the philosophy is that the role of data driven learning is maximised, and the innate information given to the agent is minimised.

[1] As must be granted, since it is manifestly not the case that children have the same competencies as adults.

[2] save, perhaps, a few reflexes necessary for survival, such as sucking and breathing.

Both these positions are extreme. Although one might argue that natural selection had led to direct encoding of world knowledge in the structures of the brain, as Churchland (1978) observes, it seems unlikely that an entirely nativist explanation can be given for our ability to form novel scientific theories — the question remains as to how it took so long for scientific theories to develop, and if they are innate, why there has been so much controversy about them. In terms of AI, a nativist explanation corresponds to the engineer encoding most of the necessary world knowledge manually within the artificially intelligent system. This is wasteful, error-prone, expensive, and has been found to be well nigh impossible.

The radically empiricist position is also wanting. In terms of natural selection, it corresponds to natural selection having encoded very little information about the world innately, presumably giving animals a very general ability to learn from experience. This is certainly false for many animals who do not have the time (or the ability) to *learn* sophisticated theories or behaviours necessary for survival empirically, and for certain human behaviours essential for survival[3], and given that evolution would seem to bias in favour of swift learning, why should it not have given organisms a nativist bias towards a successful sophisticated theory of the world it will find itself situated in?[4] In terms of AI machine engineering, this position corresponds to the engineer encoding virtually no domain specific information into an artificially intelligent system. Since the engineer almost always has at least some such knowledge, not to exploit this source of knowledge seems unnecessarily self-handicapping.

Clearly both these radical positions are extremes, and the truth about the nature/nurture controversy, as ever, lies somewhere in between, as does its utility for the approach to generating artificially intelligent systems. However, it is an interesting question in itself to find out how much *can* be achieved without recourse to nativist explanations of sophisticated theories — is it possible to find a system of theory acquisition which takes

[3]For instance, children do not need to learn how to breathe or suck, despite the fact that this takes considerable coordination. In the animal kingdom, house flies do not have to learn how to fly, and although there is some evidence that empirical information plays a role in their learning to see, evidence from flies reared in the dark suggests that within seconds of being exposed to light, they can use visual information to avoid predators (Kral & Meinertzhagen, 1989).

[4]This point is more fully discussed in relation to language acquisition in the conclusion, along with a fuller discussion of the nativist/empiricist division in language acquisition.

as input raw data from real world processes, and produces as output a theory of the processes which gave rise to the data? Enthusiasm to find an answer to this question is more than just intellectual curiosity. From a scientific, and engineering point of view, such a process would seem to have several allures.

Evolutionary Efficacy If a system could be found which generated a sophisticated theory empirically, then the need for a nativist bias would be removed or reduced, and evolutional arguments that strong innate biases are desirable would be reduced too. This is because any nativist bias is, of its nature, insensitive to the particular conditions existing in the world in which the organism will find itself situated. Consequently, the most successful theory must be at least as good as an innate one, so an evolution path which finds a sophisticated theory acquisition system will not find it helpful to provide native theories.

Cognitive Simplicity Concentrating on such an acquisition system allows a unification of study in many cognitive domains. If structure is found in sound empirically in roughly the same way that it is found in vision or language, then lessons from any domain may be usefully brought to bear on the study of cognition in any of the others.

Engineering Utility From a machine intelligence point of view, specifying highly domain specific information is a lengthy, error-prone and expensive business. That sophisticated theories of domains might be learned simply by the analysis of readily available information from these domains is attractive to anyone who wishes to build an artificially intelligent system operating within such domains.

The primary motivation for the work presented in this thesis is the last of these lures, since this thesis concentrates on techniques which might be implemented by computers (or artificial neural networks) to find structure from raw, unlabelled data. Since the general theory finding problem is clearly very difficult, this thesis concentrates on finding some of the structure which exists in natural language.

1.2 Outline of the Thesis

One of the fundamental practical problems facing any system which seeks to learn a sophisticated theory of a domain is that sophisticated theories often require the assumption of hidden entities which have an unknown relationship to the observed data from which the theory must be generated. Often the form of the representation used to represent information within a domain is crucial if a sophisticated theory of that domain is to be induced. Chapter 2 discusses issues surrounding the form of representation, the mapping between the real world and representations of the world, and what information might be available in this mapping to find structure which exists in the real world process being represented. This chapter presents a general foundation for the experiments reported later.

This thesis concentrates on finding structure in natural language texts. Chapter 3 discusses traditional approaches to the study of syntactic structure in natural language, and places the work which will later be presented in context within this body of work. Roughly, the structure which linguists postulate to be present in language (word classes, phrases, and so on) will be assumed to be the definition of *what* is to uncovered by the learning techniques introduced later. Thus, the validity of the results of the experiments presented later will be judged by traditional linguistic criteria.

Once language has been defined, we turn to how it might be learned. There has been some work done concerning the learnability of formal languages. Chapter 4 reviews this literature, and discusses its relevance to the problem of learning a natural language. An analogy is drawn between scientific theory acquisition (by the scientific community), and learning language. It is argued that this is both a more realistic model of how children learn language, and a more promising methodology for the machine acquisition of language, than the paradigms used for learning formal languages.

Chapter 5 introduces the theory of unsupervised statistical classification used in the experiments of this thesis. Statistical and linguistic motivation is given for the choices of representation of linguistic information and statistics of this representation which were made.

Chapter 6 describes several experiments designed to uncover structure in several real and artificial domains, culminating in the application of the techniques described in chapter 5 to words in natural language. A hierarchical classification of 2000 common English words is derived, and empirically assessed with respect to the linguistic concept of word classes. It is shown that highly significant groupings of determiners, singular, plural and mass nouns, adjectives, various forms of verbs, prepositions, adverbs, WH-words, conjunctions, proper nouns, and even non-word classes can be uncovered by the direct application of these methods. This chapter also shows how semantic word classes can be uncovered by utilising readily available non-linguistic information.

Chapter 7 takes the work of chapter 6 further, applying the techniques to short sequences of words, and uncovering significant structure at this level, including simple noun phrases, n-bar groups, prepositional phrases, verb forms, and even very simple sentences. The technique is then applied to sequences of these sequences, facilitating the uncovering of more complicated verb phrases and sentences. This chapter represents the state of the art in the automatic recovery of linguistic structure in an entirely unsupervised manner.

Finally, Chapter 8 details how the techniques described in this thesis might be implemented in standard neural networks, and presents a learning algorithm which can learn the representations necessary to implement this technique. Emphasis is placed on the utility of the statistical interpretation of neural networks in finding good neural implementations of structure finding algorithms.

1.3 Background

1.3.1 Zipf's Law

The first outstanding statistical work on natural language was performed by the linguist Zipf[5], who found that if the words in a large text of English were counted and

[5]Although generally attributed to Zipf, this law was noted earlier, probably first by J. B. Estroup (see Geoffrey Sampson's article on statistical linguistics in the *Oxford International Encyclopedia of Linguistics*.

ordered from the most frequent to the least frequent $(w_1, w_2, \ldots, w_n)$, then the frequencies $(f_1, f_2, \ldots f_n)$ roughly obeyed Zipf's law (Zipf, 1935), which states that $f_n \propto \frac{1}{n}$. This was later refined by Mandelbrot to a better approximation of $f_n \propto n^{-1.05}$. Others have since found that close analogues of Zipf's law apply at all levels of language, from the phoneme to the sentence (Altmann 1980; Altmann & Schibbe 1989). The Zipf-Mandelbrot law is very fortuitous for anyone seeking to uncover structure in language, since it implies that if we concentrate on only the most frequent types of words and structures in language, we shall find that a significant proportion of the corpus can be represented. For instance, if there are 100,000 English words, then Zipf's law predicts that $\frac{\log(2{,}000)}{\log(100{,}000)} = 66\%$ of almost any English corpus will fall within the first 2,000 most common word types. This has two effects. First it reduces the number of distinct structures which need to be considered in order to find a good approximation to a large proportion of the language — in the above example representing just 2% of the word types found in any corpus will allow us to represent 66% of the tokens which occur in it. Secondly, it ensures that frequent structures are frequent enough to be able to collect reliable statistics about them. Thus it is possible to apply a 'most frequent first' approach to uncovering structure.

1.3.2 Communication theory

Shannon (Shannon 1948; Shannon & Weaver 1949) considered the problem of communicating linguistic information via channels of limited capacity. In particular, he considered how simple statistical models of language could be exploited to find encodings of natural language which allowed the greatest possible amount of linguistic information to be communicated in the shortest possible time across a communications channel of fixed capacity. This work has directly given rise to the concepts of entropy (uncertainty), and perplexity, by which the quality of a language model is measured.

1.3.3 Complexity theory

A separate strand of work from the linguist Solomonoff (1964), and the computer scientist, Kolmorogov (1965), considered issues in computational complexity. They proved

that there was always a shortest program-input pair for a Universal Turing Machine (UTM) to perform a given task (such as prediction within a domain, or identifying the strings of a formal language), and that there existed a constant C such that for any two UTMs, T_1, T_2, the shortest program-input pair for T_1 differed from that for T_2 by at most C.

This has given rise to the minimum message length (Rissanen, 1978), and the minimum description length (Wallace & Boulton 1968) paradigms, which define the most compact representation of some information (such as strings of a natural language) to be the length of the shortest specification needed to make a UTM reproduce the original information. Clearly, if the strings are the output of a simple, finite grammar, then specifying this grammar specifies the corpus. Work in this paradigm has given rise to work by Wolff (1978, 1992), who showed how simple sentences could be split up into constituents using compression methods, and Ellison (1991, 1992) who showed how compression techniques can be used to find optimal finite state automata representations of phonological information. These FSAs can be interpreted linguistically as embodying certain linguistic rules, so compression can lead directly to linguistic interpretation.

1.3.4 Recent statistical work

Recently, there has been an explosion of interest in applying statistical techniques to many problems in natural language processing (NLP) (Jelinek 1986; Garside et al. 1987; Brown et al. 1988; Fujisaki et al. 1989; Huang et al. 1990; Powers 1991; Powers & Daelemans 1992; Pereira & Schabes 1992; Schabes 1992). These researchers had various motivations for wishing to pursue the statistical approach in uncovering structure. In particular, the statistical approach seems most appropriate if the task which the NLP system is assigned can be interpreted as one of prediction. For instance, in speech recognition, recognising a sequence of words is closely related (via Bayes' theorem) to the task of predicting what a sequence of words sounds like when spoken (Huang et al. 1990). The problem of assigning grammatical categories and structures to words in context (Garside et al. 1987; Fujisaki et al. 1989) is a straightforward prediction problem. Assigning a candidate translation of English text into French (Brown et al.

1988) can be thought of as predicting the French translation of words, and their order, given knowledge of the English text. In fact, it is arguable that any learning technique can be interpreted as finding generalisations which make predictions. These researchers define stochastic models which model how a language user writes text.

The problem with this approach is precisely that it relies on finding a good language model — a stochastic system which models the language users' generation of texts accurately. Given that this is an immensely complicated process, it seems unlikely that such sophisticated statistical systems will be possible to invent, or tractable to fit statistically from natural language data. Nevertheless, the question of whether a model can be found which is 'good enough' for a task (e.g. speech recognition, parse disambiguation, and so forth) remains the subject for further research at the time of writing. At the moment, the best stochastic language models (in the sense of being the most predictive) are linguistically unperspicuous insomuch as they cannot be interpreted as embodying a grammar for natural language.

The problem of prediction, however, is very closely related to the problem of finding a sophisticated theory of a domain. A sophisticated theory which closely reflects the true structure of the real-world process[6] which generates the data will be of more use than a simple theory which does not reflect this structure for making predictions, and one criterion of the utility of a sophisticated theory is how well it can be used for prediction. This thesis does not concentrate on the problem of prediction. Rather it concentrates on the problem of finding linguistically interpretable structure in language directly. However, as will be discussed in chapter 2 and chapter 5, the methods it uses to achieve this can be interpreted as ones of prediction, where structure is recovered which respects the similarity of statistical predictions which can be made about the context of occurrence of linguistic items. In other words, the fundamental approach of the learning techniques used here is that similarity should be defined between words so that words which make similar predictions of the values of various statistics are themselves defined to be similar. One may then exploit such structure to find abstract classes, and to define structures which depend on these classes in order to find further structure.

[6]Philosophical quibbles about the realist/anti-realist interpretation of hidden entities notwithstanding.

1.3.5 Connectionist work

There has been work done on language acquisition in the field of connectionism. Scholtes (1991) shows how a simple connectionist topographic mapping system (a Kohonen network) can be used to classify artificial data to reveal a topology consistent with the underlying classifications used to generate the data. He also showed that the technique looked promising for real data. Indeed, the Kohonen network is more fully discussed in chapter 8, and it is shown to be a particular example of the general statistical technique described in this thesis.

Also, there is work from Elman (1989, 1990), who trained a simple recurrent network with a prediction problem from data from the output of various simple formal systems. Since Elman's work does not concern natural language, it is of limited relevance to the work presented here, but as a potential means of providing a classification of a domain on the basis of solving a prediction problem, it has more in common with the work described above than the work to be described here.

1.3.6 Structural Linguistics

Prior to Chomsky's highly influential work *Syntactic Structures* (1957), the emphasis of work in linguistics was on specifying methods whereby an investigator could define the structure of any language using empirical, generally bottom-up, methods. Work from this paradigm includes that of Bloomfield (e.g. 1933) and Harris (e.g. 1951). The emphasis was on defining *discovery procedures*, which were algorithms which could allow the researcher to define linguistic entities (e.g. word classes and dependency relationships) automatically from hearing natural language being spoken by native speakers, and on being able to ask them questions about that language.

This paradigm was criticised by Chomsky (1957) for failing to properly dissociate the definition of what structure existed in natural language from the procedures which allowed that structure to be found, and of being too ambitious in any case, there not being enough information in a corpus of a natural language to define its structure. One way of viewing the work in this thesis is as providing statistical discovery procedures for word

and phrasal classes in natural language. More is written about the structuralist paradigm in chapter 3, but the work of many structural linguists (e.g. Halliday 1961) shows that it is certainly possible to dissociate the definition of structure in natural language from procedures which discover that structure within the structuralist paradigm, and whether there is enough information in a large corpus of natural language to discover much about its structure is an empirical matter, and the results presented here suggest that some linguistic structure is easily discovered by a simple statistical analysis.

What has made the work in this thesis possible is the availability of large corpora of natural language in machine readable form, and the availability of machines which can analyse large corpora in a variety of different ways very quickly. Perhaps it was precisely the lack of these materials which made the structuralist programme infeasible during the 1950s, rather than some fundamental theoretical flaw.

1.3.7 Most relevant work

The first attempt to apply some of the methods used by this thesis to uncovering syntactic structure in language were made by Kiss (1972) who used bigram statistics of the occurrence of words in children's stories to classify about 30 words. He found that these words clustered largely according to their syntactic category. The main motivation for this work was concerned with the psychology of language acquisition, and the results were quite impressive. However, the study was not expanded to consider whether a more complete classification of language could be achieved in this way, and no attempt was made to uncover higher level units than the word class.

Similar methods to those used in this thesis have recently been used (independently) by some researchers to derive word classes (Brill et al. 1990; Brown et al. 1990; Kneser & Ney 1991; Finch & Chater 1991; Hughes 1992). Brill (et al) applied statistical clustering to a stochastic model of language in order to find word classes, as did Kneser & Ney and Brown et al, while Hughes used a method very similar to that used in this thesis. None, as far as I am aware, have applied the technique to find higher level structure in an analogous way. Redington (1992) applied the techniques developed in this thesis to a large corpus of child language data, and showed that these techniques were capable

of uncovering significant linguistic structure in the speech of parents to children.

Recently, there has been interest in using self-organising techniques to uncover structure in natural language. This thesis is written in the spirit of the proposals expressed in Powers & Daelemans (1992) to use new technology, and the increasing availability of data, to apply large scale, exploratory, self-organising techniques to facilitate the discovery of structure in natural language and other cognitive domains. Indeed, in order to uncover hierarchical structure in chapter 7, this thesis employs a technique similar in spirit to that used by Powers (1992), where different 'levels', where levels roughly correspond to the number of words or letters in sequences being considered for classification, are defined, and linguistic structure is uncovered hierarchically where structure in each level is found in terms of regularities over the level below.

Chapter 2

Representation, Similarity and Learning

One reason why learning the structure of a domain without any prior knowledge is so difficult is that both an appropriate set of categories to describe the phenomena and the regularities defined in terms of those categories must be learned from scratch. Thus the learner must solve a 'bootstrapping' problem (e.g. Pinker 1987): the specification of a set of rules presupposes a set of categories, but the validity of a set of categories can only be assessed in the light of the utility of the set of rules that they support. *Prima facie*, at least, this implies that both rules and categories must somehow be derived together. However, the space of possible of rule/category combinations is so large that it seems unlikely that such an approach will be feasible for learning the structure of any but the simplest domains. In order to learn about a domain, it is necessary to represent items within it. The general question of representation, and its connection to computation and algorithm, is now more fully discussed.

2.1 What Represents?

Representation is an important practical issue whenever models which correspond to parts of the world, or records of the world, are needed. For instance, a tape recorder

records sound, and in so doing can be thought of as representing that sound. Often it is necessary or desirable to manipulate many different representations of a source. For instance a television picture is represented in many ways in its transfer between real world situation and sitting-room TV. First a television camera translates light into electrical signals which are then encoded into an electromagnetic signal which is transmitted. This signal is then decoded by the television receiver first into another electrical signal and then back into light by a cathode ray tube, or perhaps into magnetic variations on a video tape. All of these different media can be interestingly thought of as representing (some aspects of) the variation of light intensity originally encoded by the camera. Moreover, insomuch as the variations in light at the camera can be thought of as representing (aspects of) the events which caused them, so too can the various intermediate representations.

Thus I propose that what counts as a representation be given a relatively liberal interpretation; I do not suggest that for x to represent y[1] it is necessary that (all facts about) y be reconstructible from x, or even that a known sub-description of y be reconstructible from x[2]. I do not insist that x be of some canonical 'explicit' or 'logical' form (e.g. Fodor's 'Language of Thought' thesis (1975)). To so insist would restrict the notion of representation so much that it would become useless for the purposes presented here. All I insist about the content of 'x represents y' is that (often) interesting statements about y can be made by knowing x. We can then ask questions about the *extent* to which x represents y, and of which aspects of y are represented by x and which are not. We can, for instance, ask questions about the relationship between linguistic descriptions of television pictures and linguistic descriptions of the events which caused them.

[1] Here, x might be an electromagnetic signal, and y the light pattern hitting the camera when the picture was created, or even the situation itself which is being recorded by the camera.

[2] Although, of course, a system which can reconstruct aspects of the original signal is very desirable if y is to be considered a representation of x.

2.1.1 Formal and Substantial Representation

One distinction worth mentioning here is the 'form/substance' distinction. The distinction is founded on the observation that *substances* (electromagnetic signals, video-tape, ink and paper) can be said to represent not in themselves, but only under some known relationship between the substantial representation, and what it is a representation *of* (thus we say 'that is a picture of Sue Lawley reading the evening news' because we know what Sue Lawley looks like. That is, we know the relationship between the television picture and the events (or people) they are pictures *of*). *Formal* representations are abstractions which can be manipulated by mathematical rules to make predictions. Logical formulas are one such example, where facts and regularities which pertain between facts, are represented as a set of logical formulas. Rules of logical inference can be applied to this representation of facts to deduce further facts (a fuller description). Thus from a partial formal description of Socrates ('Socrates is a man'), and a known regularity between these logical formal descriptions ('all men are mortal'), it is possible to infer a fuller formal description ('Socrates is mortal'). Another example is that of a physicist's model of fluid. The atoms and the forces between them can be idealised, and formally represented by a set of differential equations which hold over an area of a 3-dimensional vector space. For varying boundary conditions, these equations may be solved to produce formal predictions about various properties of the representation of the fluid at various points (for instance, instantaneous velocity and rotation). Since there is a clear isomorphism between the formal description of the fluid (a volume in 3-space), and the actual substantial fluid being modelled, formal predictions can be interpreted empirically (that is, substantially in the form/substance distinction).

The representations commonly used in mathematical formalisms include vector spaces, logical formulas, real and complex numbers, functions from time to real numbers, statistical contingency tables, and so on. They are useful in so much as a set of mathematical tools is available to perform formal inference using these concepts. The validity and utility of these inferences for *substances* depends on the knowledge of the empirical link between the substance and the form, and the empirical validity of the assumptions made in making the formal inferences. Formal descriptions, therefore, will largely be

considered as a precise and convenient way of talking about approximations to substantial structures.[3] Often it will be the case that many substantial structures can be given the same (possibly idealised) formal description. For instance, all computers executing a LISP function can be formally modelled by a formal analysis of an individual computer running that LISP function. Later, we shall discuss the form of a representation, and it will be useful to consider vinyl records, magnetic tapes, and compact discs as all having the same *form* of representation of sound, albeit *implemented* in different substances.

2.2 Denotation, Representation, Computation, and Algorithm

Representation is a central issue in Cognitive Science too. In Psychology, the assertion that it makes sense to talk about 'mental representations' in the explanation of psychological data is one which separates the 'cognitivists' from the 'behaviourists', and as such is one of the foundations of the subject. Computationally, researchers working in artificial intelligence (henceforth, AI) have long realised that the computational structures used to represent a domain play a very important part in solving problems within that domain.

Marr (1982) presents a general analysis of computational devices. In this chapter, the term 'computational device' receives a very broad interpretation, encompassing not only electronic computers, brains, and the like, but *anything* which might usefully be thought of as transforming, processing, or storing information. Therefore television receivers (electromagnetic information to light information), tape recorders (sound information to magnetic information), and even motor vehicles (mechanical information (supplied by the driver) to mechanical information (e.g. about the orientation of the wheels)), can be thought of as computational devices.

Marr (1982) split theoretical analysis of the computation actually done into three 'levels'. The first was the computational level. In foundational computer science, this

[3] Precise, because the forms we shall use have precise mathematical interpretations. However, these nice mathematical properties typically lead to a model which does not precisely apply to the substantial structures being modelled, hence the use of the phrase 'approximation to substantial structures'.

corresponds most closely to the *denotational* level (e.g. Stoy, 1977), which provides a formal description of the computation in terms of the mapping between the representation of information present at the start and end of the computation. If this formal system is thought to be formal model of some real-world (substantial) events, the denotational description, therefore, can make reference to facts about the world which are external to the computational device. These might include such things as the goals and specification of what the device is to compute in terms of the external 'real' world in which it is situated, or they might be mathematical domains, as they are in denotational semantics (Stoy 1977).

The second, or 'algorithmic', level describes how the computation is performed: the processes which are used; the order in which the manipulations are performed; and so on. In the algorithmic description of a computation, there is no place for descriptions of the world outside the computational device. The algorithmic theory describes how the computation is performed in terms of the representations used internally, and does not depend on the interpretations ascribed to them in the outside world.

The third, or 'implementational' level describes, as one would expect, how the algorithm is *implemented* in the actual hardware of the device.

Most important to this thesis are the first two levels, and the most important insight Marr's analysis of computation gives is the fact that while the computational level explanation can refer to facts which are external to the device, the algorithmic level explanation can refer only to abstractions of events wholly within the device. Representation is common to both levels of explanation: for the computational explanation, representations are *of* facts about the external world, while for the algorithmic explanation representations are what is manipulated and transformed during the process of computation. Yet algorithmic and computational descriptions describe the *same* device. One question worth asking is whether the distinction is valuable in the analysis of computational devices. Conceptually, the distinction between computation and algorithm is very valuable, and practically it has given rise to a large field of research in foundational computer science (Stoy 1977; Scott 1981), but it is also a very real distinction, since while any one computational description can be computed by many algorithms, and any

algorithm can satisfy many computational level specifications, some algorithms are not consonant with some computations. Two examples will be given to illustrate this point.

2.2.1 Representation in Denotational and Operational Theories

Representation plays two roles in theories of computational devices. On one hand it plays a role in the *denotation* of what the device computes: what is the external relevance of what is computed? On the other hand, representation is that which is manipulated, or *operated on* by computational systems entirely internal to the computational device: how are representations manipulated during the operation of the algorithm?

As a simple example of the dependency between these two questions, let us consider Marr's own example of a till, such as that used in supermarkets. Denotationally, this till performs addition over the positive integers, and as such, this might be a formal model of the calculation of the amount of money a customer owes the super-market for the goods they have bought. The denotation might be from a set of keystrokes (corresponding to the till-operator keying in the prices), which might be further denoted as a sequence of numbers, $\{x_i | i = 1, 2, \ldots, n\}$ to a number, which is supposed to represent the total bill, and has denotation $\sum_{i=1}^{n} x_i$. The *algorithmic* or *operational* theory describes how a particular set of *representations* of numbers, generated by keystrokes, is transformed to a number (or, rather, some binary representation of a number) which can then be displayed in decimal form.

It is clear that there are many distinct algorithms capable of performing addition. However, what is not so obvious at first sight is that this same algorithm also computes a large number of other denotations, both in (substantial) terms of sequences of keystrokes, and in (formal) terms of functions between numbers. To see this, simply permute the numbers on the keys of the till. Now, the function computed by the till is no longer $\sum_{i=1}^{n} x_i$, but is rather $\sum_{i=1}^{n} F(x_i)$, where $F(\sum_{j=0}^{n_i} 10^j k_{ij}) = \sum_{j=0}^{n_i} 10^j f(k_{ij})$, where k_{ij} is the jth decimal digit of the number x_i, and $f(.)$ is the permutation of the digits described above. The point is that the computational level description of the till has changed, but since nothing internal to the till has changed, the algorithmic level description has not. However, the resultant denotation is a poor model of what the customer owes the

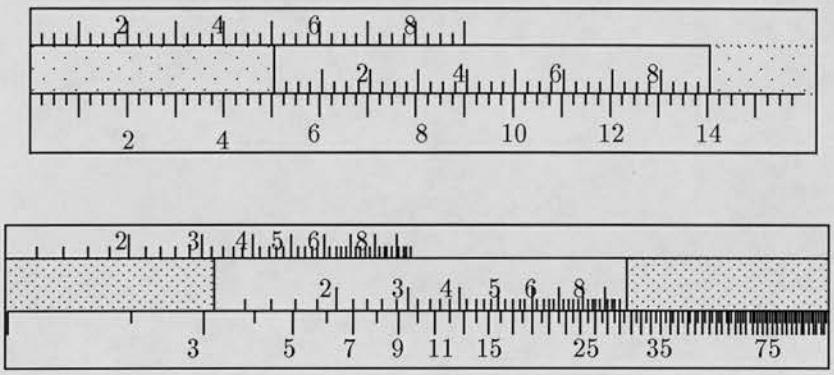


Figure 2.1: Two slide rules. The upper one set up to do addition, the lower one is equipped with logarithmic scales to perform multiplication.

supermarket, and so the till does not now satisfy the goal of the computation.

Perhaps an even better example to illustrate the difference between representation to the denotational (computational) and algorithmic (operational) theories of a computational device is the slide rule. A slide rule consists of two parts: a base, which is rather like a ruler, and a sliding carriage. The algorithm for operating a slide rule is as follows:

START

> Find the (representation of the) first number on the top scale.
>
> Move the carriage so that its left side is flush with the first number.
>
> Find the (representation of the) second number on the scale on the carriage.
>
> Read the answer off from the lower scale.

END

A slide rule can easily perform addition: to add two numbers together, simply slide the carriage along the rule until the left hand edge of the carriage is parallel with the mark on the scale corresponding to the *denotation* of the first number, and read off the denotation of the second number from the carriage. The mark on the long scale is (approximately) the denotation of the sum of the two numbers (see figure 2.1).

However, by changing the denotation of the scales, the *same* algorithm can perform multiplication rather than addition. One simply changes the relationship between length

and the denotation of numbers to be logarithmic rather than linear, and the same algorithm now can be used to perform a quite different computation which, unlike the example of the till above, is useful and might well be the goal of a practical computational device. Moreover, by varying the three scales, one can define a general class of slide rules which all compute different arithmetic functions using the same algorithm.

However, there are some things which can be said about the denotation of the function being computed merely from knowing the algorithm which computes it. In the case of the till with permuted buttons, it is still the case that $x + y{=}y + x$, and that since the number zero plays a certain denotational role in addition, it is still the case that $\exists x(\forall y, x + y = y)$, and other regularities due to the fact that the till has an algorithm which *can* be interpreted as computing addition given a suitable interpretation of its input.

In the case of the slide rule, we note that this analogue computer only *approximately* calculates the function which has been ascribed to it. There are several reasons for uncertainty in the function it computes: firstly, it is not possible to find a position of the scale which *precisely* corresponds to the number we wish to represent. Secondly, it is not possible to precisely line up the carriage with an arbitrary chosen position on the top scale of the slide rule. Thirdly, it is not possible to interpret a position on the answer scale as a precise number — there will always be errors due to measurement, so we cannot find the denotation from the representation of the answer. Fourthly, it is not possible to construct a precise scale anyway, and deviations in the scale from being truly logarithmic or linear with distance may lead to a different function being computed than the one intended. As a consequence of this, the same calculation performed many times may yield different answers, and the same answer may sometimes be obtained from calculations which are known to be denotationally distinct. Note that there are two classes of reason for this uncertainty: firstly, uncertainty of representation with respect to denotation (interpreting a number as a point on the scale, and vice-versa (measurement), and also the integrity of the scales with respect to the denotation of the function to be computed), and secondly, uncertainty due to the fact that the carriage cannot in practice be moved to every point, and that the position of the left hand edge

of the carriage might not precisely determine the position of the rest of the carriage (due to physical forces of friction causing slight warping of the material of the slide rule. This is especially true of slide rules made of an inappropriate material, such as plasticine, for instance).

2.3 The Marrian Model of Computation

The analysis of computation we have been investigating, due to Marr (1982), is illustrated in figure 2.2. A computation can be viewed as a mapping between substantial representations at the implementational level, as a mapping between formal representations at the algorithmic level, or as a mapping between denotations (real-world) at the computational level. The relationship between the denotations of the items of interest in the world being represented, and the representation ascribed them (by some process I shall call an *intentional operator*), is often not deterministic. The relationship between representations and the denotations ascribed to them is via a mapping I shall call the *extensional operator*. Again, this need not be deterministic. Individuals and situations which are in all important respects alike might (and do) give rise to different representations of them, and, since in many practical domains such as sound and vision there are many more distinguishable sounds and pictures than there are distinct representations in any discrete computational system, the same representation can be caused by situations which are different, and so might receive many possible denotations.

In the case of the till presented above, $\mathbf{D}_1$ is the set of prices of the goods the customer bought, formally modelled as a set of numbers, represented as a sequence of binary coded numbers in $\mathbf{R}_1$. F is the sum of the numbers, and f is an algorithm which performs binary addition over a set of numbers. $\mathbf{M}_1$ might be variations in potential in some transistors, while $\mathbf{M}_2$ might be patterns on a light-emitting diode display. $\mathbf{R}_2$ might be a binary encoding of a number, while $\mathbf{D}_2$ will be the space of total values the customer owes the supermarket, modelled by a real number. It is easy to see that just changing $\mathcal{I}_1$ has the effect described above of completely changing the denotation, F, of what the till computes, while leaving everything else alone.

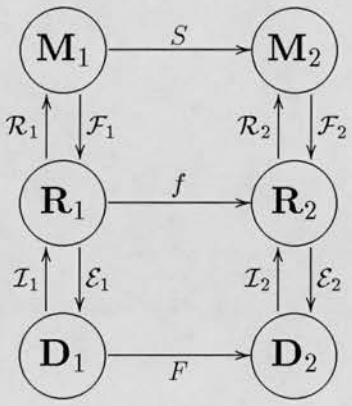


Figure 2.2: The model of computation this chapter considers. A computation can be considered denotationally as a function between (formal models of) aspects of the world, $\mathbf{D_1}$ and $\mathbf{D_2}$, or as an operation or transformation of representations ($\mathbf{R}_1$ and $\mathbf{R}_2$). The mapping between denotations and representations is via an *intensional operator*, here denoted $\mathcal{I}$. The mapping between representations and extensions is by way of an *extensional operator*, here denoted by $\mathcal{E}$. This representations will be implemented in physical media $\mathbf{M}_1$ and $\mathbf{M}_2$, and the *substantial operator*, $\mathcal{R}$ maps between representational form and representational substance, while the formal operator, $\mathcal{F}$, maps between representational substance and representational form.

In the case of the slide-rule used for multiplication, $\mathbf{D_1}$ is the space of pairs of numbers between 1 and 10, $\mathbf{R_1}$ is a formal model of the space of pairs of lengths on the top part of the rule, and the carriage, while $\mathbf{M}_1$ is set of possible substantial states of the sliderule. $\mathbf{R_2}$ is a formal model of the lengths in the bottom part of the slide-rule, while $\mathbf{M}_2$ is again the set of states of the sliderule. $\mathbf{D_2}$ is the space of real numbers between 1 and 100. $\mathcal{I}_1$ is the mapping between pairs of numbers and pairs of lengths given by the upper two scales, while $\mathcal{E}_2$ is the mapping between lengths and numbers given by the lower scale. $\mathcal{R}_1$ represents how to set the sliderule up given knowledge of the two lengths, while $\mathcal{F}_2$ is the mapping between states of the sliderule and lengths on the lower scale. Changing the slide rule from one which calculates sums to one which calculates products requires that we change both $\mathcal{I}_1$ and $\mathcal{E}_2$. The observation that measurement is uncertain is explained by non-determinism in the mapping $\mathcal{F}_2$ (or even $\mathcal{E}_2$, if measurement error is accounted for by incomplete knowledge of the mapping between lengths and numbers). The observation that the scales are impossible to make completely accurate is accounted for by impossibility of completely defining the mappings between the algorithmic and implementational levels. The observation

that an inappropriate material for the sliderule, such as plasticine, produces unreliable results is accounted for by the fact that the maps between abstract lengths (**R**) and substantial states (**M**) is unstable and changes with time.

2.4 Representational Media and Form in Marr's Model

The form and medium chosen to represent information to a computational device is now considered (the domains **R** and **M**). For the purposes of this thesis, we shall be especially concerned with representing text in a way which facilitates the acquisition of theories of textual structure (linguistic theories of syntax). However, the discussion will be kept at a relatively general level.

Examples of representational media in use today abound: ink and paper; vinyl records; magnetic tape; compact disks; silicon; photo-sensitive paper; electromagnetic waves; and so on. These can be used to store many different *forms* of information.

For example, a vinyl record stores the form of the variations in air pressure caused by sound (within certain frequencies and volumes) by variations in the height of its groove[4]. A cassette records the same form of information by variations in the magnetic orientation of particles on a magnetic tape. A compact disc stores the same form of information by a sequence of numbers obtained directly from an electrical transduction of the position of the microphone which recorded the sound at various times during the sound's duration. All three media have in common that they facilitate an approximate recreation of the sound by using reverse transduction devices, and that these transduction devices have, at least at some level of form, similar algorithmic descriptions, this being due to the fact that it is necessary to recreate an electrical signal which is, in the case of a record, proportional to the displacement of the groove; in the case of the magnetic tape, proportional to the deflection of the magnetic orientation; and in the case of the compact-disc, proportional to the number stored on disc. All these repre-

[4]The actual transduction of sound to vinyl is made via an electrical device which transduces sound into electrical variations, and thence into mechanical variations. Consequently, the information which is actually stored is of a mechanical transduction of an electrical transduction of a mechanical transduction of the sound. In line with the comments in the introduction to this chapter, I shall simply say it is a representation of the sound, since interesting facts about the sound can be retrieved by knowing the state of the record which records it.

sentations of sound have a more or less direct mapping to a signal which is formally represented as a real function of time.

But it is possible to record sound using representation with a very different form. Printed music is one such example. That this is a different form of representation is evidenced by the fact that is is most naturally thought of as being a means of describing how to recreate the sound not by directly recreating an electrical signal as in the previous form, but rather by using an instrument (such as a piano) to create the sound. To this end, the representation can readily be formally decomposed into, for instance, a sequence of notes, which are themselves arranged in 'phrases', which must be played in a certain 'mood' at a certain volume, and so on. Thus the algorithm used to recreate the sound from the printed music is in some sense *formally* different from the algorithm which recreate sound from a direct representation of the signal.

The question now presents itself as to what is the best representational form for a particular task, and what is the best representational medium for storing and manipulating this form? The answer to the first question depends closely on the computational description of the task, including the goals, the form of the available data, and the algorithms which can be defined to perform this computation. The answer to the second question depends largely on the substantial implementation of the algorithm, and the properties we wish $\mathcal{R}, \mathcal{F}$ and S to have, such as resistance to change with time, accuracy (very little non-determinacy), and so on.

2.4.1 Good Representational Forms

The first task we have is to choose an initial representational form for the domain with which we are concerned. This is the structural backbone of any theory which might be derived or posited. For example, one form for the English language might be strings of letters (letters being some formal abstraction which covers many media: ink blobs on paper, pixels on a video display unit, electronic variations corresponding to the ascii representation of letters in a computer's memory, and so on). For the spoken language it might be a function of time representing the sound signal (or magnetic variations on a ferrous tape, or undulations in the grooves of a record, and so on). Experience

has shown that choice of representational form is crucial to how a theory of a domain develops.

Choosing a representational form for developing a theory of a domain is, clearly, not straightforward. From an a-priori point of view, and by looking at what properties representational forms of many sorts possess, there seem to be several desirable properties for such a representational medium. Some of these are listed below. The phrase 'items of interest', although clearly unwieldy, should be taken to refer to what we wish to represent.

In the following, let $\mathbf{D}$ be the set of entities in the world we wish to represent by elements of a formal representational domain $\mathbf{R}$, implemented in some medium which has a set of states $\mathbf{M}$. Suppose we have $d \in \mathbf{D}$, then its formal representation will be written $\mathcal{I}(d)$, which is a member of $\mathbf{R}$, and the state of the medium will be $\mathcal{R}(\mathcal{I}(d))$, which is a member of $\mathbf{M}$.

Availability It should be possible to easily translate the items of interest into their appropriate representation. In the case of the magnetic tape referred to in the first paragraph, such a translator is a cassette recorder. In the case of written language, it is the brain[5]. This criterion concerns the map between the *denotation* and the *representation* of information — If the algorithm is to perform a certain denotational function, then it should be possible to readily represent denotations.

This corresponds to insisting that $\mathcal{I}$ and $\mathcal{R}$ be relatively easy to calculate.

Expressiveness One of the roles of representational form is to make distinctions between states of affairs which pertain in the world (at the denotational level) and to mirror the identities and similarities which exist there. Thus for a function from time to the real numbers (sound waves) to be a good formal representation of sound, different sounds would have to give rise to different formal representations of them — different sound-waves. Thus there must be enough distinct formal representations to be able to capture all the distinctions which we wish to represent. Thus, for example, a two-valued variable is certainly not *expressive* enough as a

[5]With a little help, perhaps, from a pen or computer.

formal representation of sound, since there are far more than two sounds we wish to distinguish. Thus we wish that $\mathcal{I}(d) \neq \mathcal{I}(d')$ whenever $d \neq d'$, or, as shall be explained below, that this is probably true.

The formal representation must be implemented in a substantial medium, such as a record or a cassette in the case of sound. The substantial medium must also be expressive enough to represent the formal distinctions we wish to make. Of course, this is not always possible since while formal representations might admit an infinite number of distinctions, it is not possible to find media which do. This gives rise to a non-deterministic mapping between formal representation, and the implementation of that representation — In general, one formal representation might be implemented by any one of a number of states in the substantial medium, while any measurement of that medium might be consistent with the medium representing any one of a set of formal representation. In terms of the sound wave and vinyl records, this amounts to conceding that any one sound might give rise to a number of distinct records[6], and that any one record could have been the result of the recording of many possible distinguishable sounds. Thus we wish that $\mathcal{R}(r) \neq \mathcal{R}(r')$ whenever the representations r and r' are not identical, but this is not always possible.

An example of failure of the *expressiveness* criterion is that of the electromagnetic representation of television pictures. Without going into too much detail, a television picture is represented by two signals, one representing the variation in intensity of parts of a picture, the other representing variations in colour of parts of the picture. These two signals are combined (the details of this process are immaterial), and this combined signal encoded as an electromagnetic signal, and transmitted. Sometimes, the variation of intensity is of such a form that the combination of these signals does not determine the original two components, since in general many colour signal/intensity signal pairs can give rise to the same combined signal. When this is the case, the television receiver might decode the

[6]for instance, due (a) to measurement errors of the sound-wave in the record construction device, (b) to the the presence of noise in the mapping between the electrical representation of that sound-wave and the mechanical variations in the height of the groove, again in the record construction device, and (c) the fact that the shape of the groove changes with time in unknown ways due to physical and chemical processes.

electromagnetic signal incorrectly, and not reproduce the original intensity and colour signals which were combined. This is the case, for instance, when a television picture of a slowly moving thinly striped shirt is taken — intensity information is incorrectly decoded as colour information. The failure of expressiveness here is therefore due to the combined signal not determining the colour and intensity signals. This is a failure in the *form* of representation, since we have two distinct possible pictures, d_1 and d_2, such that $\mathcal{I}(d_1) = \mathcal{I}(d_2)$. We also note that in this case, the pictures are not very similar to each other.

Operational Utility The representational *form* should be convenient for performing useful manipulations. In the case of the formal representation of sound as a sound-wave, the fact that parts of the representation of a sound can be simply (and formally) mapped onto representations of new sounds, allows valid representations of new sounds (e.g. interval parts of other sounds) to be defined from representations of old sounds (so, for instance, a single track on an album can be represented as a sound-wave if we have a sound-wave representation for the entire album, since the track is an interval part of the entire album). The implementation of this representation in substance should allow a (substantial) system to use the information to perform useful tasks (to continue the example, disk jockeys can use vinyl albums to play single tracks in discotheques using a record player).

On the other hand, the printed representation of music allows a pianist not only to play parts of the musical piece, but to vary the speed, the mood, the tone, and the loudness of the piece, but the penalty paid for such a sophisticated representation is that in order to play music one needs a far more sophisticated computation than is needed to replay music in a tape recorder.

In a knowledge based reasoning system, the manipulations to be performed are inferences and predictions about the state of 'knowledge' represented. Propositional representation forms allow the tools of logic to be naturally used to make inferences in such domains. Propositional representations allow a close analogy between denotational and algorithmic form, denotation being the *meaning* of a proposition, operation being a *proof* of that proposition.

Researchers have found that this approach, however, has serious problems of tractability in making inferences from large knowledge bases, that propositional information is far from readily available about the world, and that the regularities between propositions which do exist are often not easily modelled denotationally by the tools of classical logic (McDermott & Doyle 1980; Pearl 1988). Consequently, although this is a formal representation with a neat formal theory linking denotational and algorithmic levels, implementational considerations (the criterion that the computation should be fast), considerations of the availability of information (how to translate real-world situations and (especially) regularities into propositional form) and the lack of an adequate denotational theory linking propositions with each other have led to a large problem in finding tractable AI systems, especially ones which have a broad coverage of knowledge (e.g. Chater & Oaksford, 1990).

To the connectionist, representations in a continuous (or discrete) vector space allow connectionist computational tools (algorithms) to be easily applied to learn and generalise a mapping from examples, or to generate clusters of points which represent similar entities, or share 'features'. The ability to generalise from a set of examples (learn) is due to the presence of statistical regularities between example tokens of the mapping to be learned *with respect to the chosen representation.* Thus the degree of success a network has in modeling a particular mapping depends on the nature of the mapping, and the way the tokens of the mapping are represented. Practically, the question of how best to represent data so that statistical or neural network techniques can be successfully applied is highly complex and falls under the heading of 'preprocessing'. An image might be represented as an array of pixels, or approximated by Chebychev polynomials, and the first few coefficients of these be used to represent the picture. For a sound wave, a direct (record-like) representation of the sound-wave is usually inappropriate for connectionist techniques, the value of the wave at a particular time ('now') being largely uninformative about interesting variations in its value at other times. A Fourier analysis of the wave, which represents the wave as a frequency spectrum, can be shown to produce a representation which is far more useful for (for in-

stance) speech recognition applications than the raw speech signal as would be recorded by a record (e.g. Huang et al, 1990), because many more regularities about the sounds of speech are simply expressible (and hence learnable by simple connectionist and statistical systems) in this representational form than in the sound-wave form.

Thus representational form, computation, algorithm, and implementation must all be consonant in a formal description of a computational device. Constraining any one (often) imposes some (complicated) constraints on all the others. Yet, there are still further constraints to be added to this analysis of computation, and these are due to the presence of 'similarity' between denotations of representations, similarity of the roles particular representations play in the algorithm which implements the computation, and similarity imposed by the non-determinism of the mapping between denotations and representations.

2.5 Similarity

Alongside the analysis of the algorithm which implements a computation, the denotation of that computation, and the representations involved in a particular computation, one can also consider an analysis of *similarity* between computations, denotations, algorithms, and representations.

The philosophical and psychological literature has extensively studied similarity (Wittgenstein, 1953; Rosch, 1973, 1975; Ortony, 1979; Tversky & Gati, 1978; Osherson & Smith, 1981; Smith & Medin, 1981). The general conclusion is that similarity is a function not only of the items in question, but also of the particular perspective from which they are viewed (or their *function* in the situations in which participate). For instance, a yellow cup and a blue cup might be considered very similar for the purpose of 'having a cup of coffee', while not if trying to find a particular cup of known colour. In any case, a well-motivated definition of similarity is elusive.

If the same item is sometimes represented in different ways, then if all representations of the item in question are to be identified, this has implications for a notion of similarity

over *representations*. As a primary analysis, let us examine three possible criteria for judging two representations as similar:

Denotationally Imposed Similarity: Suppose there is a well-defined real-world notion of similarity between denotations. Then two representations can be held to be similar if their denotations are similar. That is, a similarity defined over denotations imposes a similarity over representations. We shall call this *denotational similarity.*

In general, well-defined measures of denotational similarity are elusive. All cups are similar if the task of a computation is to make a cup of coffee, but not is the task is to find all the red objects in a room. All written tokens of the letter 'a' are similar too for an optical character reader, but not if the task is to classify documents according to whether they are type-written or hand-written. In general a full account of denotational similarity is highly complicated, but for some denotational spaces well-defined mathematical or statistical notions of similarity are available (e.g. $|x - y|, x, y \in \mathbf{R}$ defines a similarity metric over the real numbers which is very useful when the real numbers are used to represent a continuous quantity (e.g. weight)).

This amounts to assuming that a metric space over denotations exists, and that the intensional operator, $\mathcal{I}$, is *continuous.*

Functional Similarity Often it is the case that the regularities which exist in the world are continuous. That is, we end up with a notion of similarity where when a computation is performed, similar individuals give similar results. For instance, addition is continuous, since if x' is very close to x, and y' is very close to y, then $x + y$ will be very close to $x' + y'$. Sometimes it is possible to assume that the function we are computing is continuous. Consequently, if the computation is modelled as a function, f, between one domain with known similarity structure ($\mathbf{R}_1$), and another domain with an unknown similarity structure ($\mathbf{R}_2$), then in much the same way as denotational similarity above, from the fact that x and y are similar in $\mathbf{R}_1$, we can infer that $f(x)$ and $f(y)$ are similar in $\mathbf{R}_2$. If f is a bijection, then it is possible to infer similarity the other way round, too, where

the structure of $\mathbf{R}_2$ is known, but the structure of $\mathbf{R}_1$ is not.

Non-Determinism of Representation: Often it is the case that the same denotation can be represented by many elements of a representational space, and one item of a representational space can be a representation of many elements of the denotational space. We shall be particularly interested in situations where the map between denotation and representation is non-deterministic, and can be subjected to analysis as a random process.

Examples of this abound. When a physicist performs an experiment, he measures various quantities. This process of measurement is a process of representation of the salient aspects of the process under study, and is subject to error and variation for many unquantifiable reasons (defects in the measuring apparatus, human error, slight changes in experimental conditions, and so on). The result of this is that the same experiment can (and does) give rise to very many different sets of results (representations of the real world event), and these sets of results should be considered similar. It is here that formalisms from statistics are applied, these being able to provide a well-defined measurement of the similarity of the data observed to some hypothesis provided by the physicist about the data, such as that it lies in a straight line.

Suppose we have a discrete set of representations, $\mathbf{R} = \{r_1, r_2, \ldots, r_n\}$, which are system-internal representations of a set of external individuals, henceforth denoted by $\mathbf{D}$, and an intensional mapping between $\mathbf{D}$ and $\mathbf{R}$ which is denoted by $\mathcal{I}(.)$. For any individual $d \in \mathbf{D}$, the representation, $\mathcal{I}(d) \in \mathbf{R}$, when repeatedly sampled, will induce a probability distribution over $\mathbf{R}$. Similarity between such probability distributions for different elements of the denotational space, $\mathbf{D}$, is amenable to statistical analysis, so a similarity measure over the set of denotations, $\mathbf{D}$, can be inferred by the functional similarity criterion discussed above. Various extensions of this observation will be the foundation of the methodology used by this thesis in uncovering similarity between linguistic units.

The power of non-deterministic similarity is that it gives us a way of inferring similarity from distinction. It embodies the observation that x is similar to y just

in case x frequently gives rise to the *same* measurements as y. Thus we need know no more about the topology of the measurement space than it can have the discrete topology — i.e. a flat set of distinguishable elements — which all sets can be given. Thus it truly is a knowledge-free technique.

From the point of view of a system which learns, non-deterministic similarity together with functional and denotational similarity can be exploited to define similarity between elements of a representational domain when we have two or more distinct representations of a denotational domain. It is very important to the process of learning to be able to determine the topology of the representational space, since so often the regularities we wish to model are continuous with respect to this topology, and consequently can be more readily learned if the topology is known. The general question of learning is now discussed.

2.6 Learning

The general problems in learning are very similar to the problems in search. Learning problems are usually cast in a form which involves finding some 'best' explanation of a set of examples out of a space of possible explanations. Consequently, constraints that help in search tasks might also be expected to help in learning tasks. First, I will outline two closely related paradigms for learning.

2.6.1 The Statistical Paradigm for Learning

The parametric statistics paradigm for learning concentrates on developing parameter fitting rules for classes of parameterised functions.

The idea is that the set of possible mappings between representations is of a certain form. For instance, if the representational spaces are vector spaces of n dimensions, the mapping is assumed to be in a parameterised class of functions, $\{F(\phi)|\phi \in \Phi\}$[7]. It is assumed that some element from this class will be at least a good approximation to the

[7]Thus Φ is a set of parameters, not the domain of F.

denotational function which is to be modelled.

The statistical task is to find the value $\phi \in \Phi$ which makes the resultant mapping $F(\phi)$ the most likely explanation of the examples seen so far, usually by reference to one of two criteria: a Bayesian one, which assumes an initial preference over values of ϕ (a *prior distribution*), and re-estimates this preference in the light of the training examples using Bayes' rule, and the MAXIMUM LIKELIHOOD criterion, which chooses that ϕ which, out of all possible values of $\phi \in \Phi$ would have made $F(\phi)$ most likely to have generated the examples.

In this paradigm, generalisation arises from the fact that choosing a particular parameter value ϕ automatically imposes a mapping over the entire space. The validity of this generalisation can be tested by seeing how well the derived function predicts new data. If it predicts new data well enough, then the function $F(\phi)$ might be accepted as a good model of the map between $\mathbf{R_1}$ and $\mathbf{R_2}$. If it is a bad mapping, this is because either (a) there is no value in Φ which reasonably models the mapping $\mathbf{D_1} \rightarrow \mathbf{D_2}$, or (b) we have found a value in Φ which models the data so far seen, but we have not seen enough data to be able to make adequate generalisations from. Problems of type (b) are possibly solved by increasing the number of examples, while problems of type (a) can be solved by choosing another, usually more complicated, set of parameterised functions.

Statistical fitting works best when the size of the parameter space is small (i.e. there are relatively few functions). This is because the problem of finding the best parameter is, in general, a difficult search problem which is most easily solved when the search space is small.

Practically, being able to implement a statistical model relies on assuming the $F(\phi)$ to be non-deterministic mappings, so that we can assign a non-zero probability that $F(\phi)$ generated a particular set of examples for almost every value of $\phi \in \Phi$. It also relies on Φ not being too 'large' a space, both so that the search problem of finding the best value of ϕ given some training examples is not too hard, and so that not too many training examples are needed to get adequate generalisation.

2.6.2 The Neural Network Paradigm for Learning

The neural network (NN) approach to learning is similar to the statistical approach to learning, except that rather than choose the value $\phi \in \Phi$ which maximises the probability of the training examples, one chooses the value $\phi \in \Phi$ which minimises some 'error function' over the training examples, and hopes that this produces a function which generalises well to new examples. Typical examples of this paradigm include the perceptron learning rule (McCulloch & Pitts, 1943; Rosenblatt, 1959), which fits the parameters of a linear threshold function, and multilayer backpropagation networks (Rumelhart & McClelland, 1986), which fit parameters to minimise an error measure using a hill-climbing algorithm.

Some work on neural networks concentrates on the problem of expanding and contracting the parameter space so that the function learned is accurate, while adequately generalising in a statistical sense (e.g. Mackay, 1991). Other work concentrates on replacing the 'error function' by one which admits a natural statistical interpretation for the function of neural networks (e.g. Spackman, 1991). More work along this line will be presented later (chapter 8).

Perhaps the main difference in approach between the straightforwardly statististical one, and the neural nets paradigm is the latter's concentration on algorithmic form, and the former's concentration on denotational form. Statistics provides computational level models of the set of data which is to be modelled, and concentrates on finding algorithms to choose the best model of the data, while the neural networks paradigm concentrates on defining a set of algorithms (neural networks), giving an algorithmic description of weight update rules and network function. Thus while a statistician might say 'let's assume the data set was generated by a joint normal distribution', the neural net engineer would say 'let's throw this data set at a net with 10 hidden units trained by backpropagation with Euclidean error function'. Nevertheless, the two approaches are closely linked, and neural nets often have statistical interpretations (Golden 1987).

2.6.3 What Helps Learning?

The above two learning paradigms have in common that they work by constraining the number of possible answers to the question 'What function models this data?' Both paradigms also have in common that they can assign 'goodness' ratings between any set of training examples and any putative modelling function corresponding to a measure of how well the putative function models the data. This facilitates gradient driven search techniques such as hill-climbing (e.g. back-propagation in neural networks, the EM algorithm (Baum et al, 1970) for certain stochastic processes), or stochastic search techniques (e.g. the Boltzmann machine (Hopfield 1982; Hinton et al. 1984), or Monte Carlo simulation (recently reborn in the Physics community under the name of 'simulated annealing') (e.g. Hammersley & Handscomb 1965; Geman & Geman 1984)). Thus, ways of translating the general computational model given above into one which can be tackled using standard learning techniques might be expected to make the task of learning denotational structure tractable, or at least give a means of attacking the problem.

The most straightforward way this can be achieved is by choosing the intensional operator ($\mathcal{I}$) so that denotations are represented in some vector space, and to define the set of possible modeling functions, $\{F(\phi)|\phi \in \Phi\}$, to be a class of mathematically or statistically perspicuous functions between vector spaces. In the terminology of neural nets, this is the *encoding problem* — how are the data encoded? In the terminology of this chapter, this problem is one of defining the intensional operator, $\mathcal{I}$, between the denotational (outside) and representational (inside) domains.

Representing denotations in a vector space offers the advantage of the easy applicability of a large class of functions which can be learned by statistical and gradient based search techniques. How well these search techniques work depends on the geometry of the search space, and this itself is contingent on the nature of the statistical regularities in the training data. If there are strong statistical regularities within a small priorly defined space of possible regularities to which the search algorithm is sensitive, then learning these regularities will be swift and efficient. If there are a large number of spurious regularities (due, for instance, to having too few training examples), or the

search algorithm is not sensitive to the regularities which exist in the training examples, or if the search space is too large or does not contain appropriate functions, then learning will be unsuccessful, or if it is successful, it will be slow.

Often, however, there is no natural mapping between the denotational domain and a vector space. If this is the case, then either techniques have to be found to impose such a mapping, or some other model of learning must be used which is not based on manipulating vector representations. I shall consider both these possibilities now in the light of the general computational model.

Imposing a vector mapping

If we are given a denotational space, $\mathbf{D}$, and wish to represent this domain in some vector space, $\mathbf{R}$, so that the available techniques of learning can be applied, then the problem is one of defining a suitable intension operator, $\mathcal{I}$, so that this might be achieved. There are many ways this can be done. For instance, if $\mathbf{D}$ is the space of symbols from some alphabet, $\mathcal{A}$, then one can re-interpret this as a vector, simply by assigning the intensional mapping between $\mathbf{D}$ and $\mathbf{R}$ by $\mathcal{I}(s) = \mathbf{e}_{F(s)}$, where $s \in \mathcal{A}$, $\mathbf{e}_i$ is a basis vector with all components 0 except for component i which is 1, and $F(.)$ is a mapping between symbols and the natural numbers which can be thought of as assigning a dimension of a vector space to each distinct symbol. This is variously called the *punctate*, *discrete*, *localist*, or *1 of N* mapping, where each distinct element of $\mathbf{D}$ is assigned a dimension in a vector space.

This mapping, however, is only feasible when the denotational space is small (otherwise huge vectors would make any search task of functions defined between them too liable to overfitting, or too large, or both), and, moreover, does not (usually) respect the similarity criteria discussed earlier. If some pairs of symbols are more similar to each other than others in terms of the function they play in theories of the data, then a better mapping might be one which respects this similarity.

One other way of forcing a representation in a vector space is to use statistical distributions of dependent representations. Much of this thesis addresses this question, but the

basic strategy is to define a similarity measure between representations on the basis of some regularities present in the domain being represented. These similarities can then be used to provide a vector-space representation for the domain, either because a vector space representation is constructed in the process of finding functional similarities between the items of interest, or because there exist many statistical procedures to convert observed similarities to a vector space representation in which Euclidean distance between representations (vector space similarity) mirrors the observed (statistical) similarities (for instance, multi-dimensional scaling (e.g. Bechtel, 1976)). A method for finding functional similarity will be outlined below in section 2.7.1.

Learning without vector spaces

Often it is the case that what is to be learnt cannot naturally be considered as a mapping between vector spaces. For instance, formal language theory considers symbolic grammars, and operations such as recursion, so central to the analysis of these formalisms, are not naturally represented in vector spaces. Finite state automata and Turing machines, also, are not naturally considered as operations over a vector space. And yet these formal systems are theoretically valuable in the analysis of real-world data (such as natural language), so it is interesting to ask what the learning theory is for formalisms which do not directly involve vector-space representations.

In natural language, some results due to Gold (1967) will be given in chapter 4 which considers search procedures through spaces of (symbolic) grammars. Solomonoff (1964) proposed a definition of the 'best' grammar for a corpus in terms of *minimal message length* theory. Gradient descent algorithms are available to search for such 'best' grammars within limited domains. For instance, work has been done in phonology using stochastic gradient descent search algorithms to search through a space of entirely symbolic models to minimise measures of symbolic complexity (Ellison, 1991; 1992). Gradient descent algorithms are also able to fit (stochastic) regular and context free grammars to corpora (Baum et al. 1970; Jelinek 1986; Fujisaki et al. 1989; Shabes, 1992). Consequently, models of learning are not restricted to learning mappings between vector spaces, and frequently there are more appropriate representational forms for data than

vector spaces.

2.7 Learning Topological Structure

Topology is the study of structure of various mathematical representational spaces. Examples of distinct topological spaces include multi-dimensional Euclidean spaces, spheres, trees, and the like. In fact, almost all mathematical spaces can be studied by some branch of topology. Some topological spaces are metric spaces. A metric space is a set of points, X, together with a distance measure, d, between pairs of points to the non-negative real numbers satisfying certain axioms. The distance metric of such spaces characterises the space in that any other space which can be interpreted as having the same distance metric is *isomorphic* to the original space, and hence is not distinguished topologically from it. Examples of possible topological spaces for perception include discs (e.g. the retina) and spheres (possibly with holes) (e.g. our skin). So, finding a metric for our representational space allows us to uncover the topology (and the geometry) of the representational space.

For all forms of learning, knowing what the topology of the representational space is in terms of the similarities which pertain within it might be expected to be very useful. It is always difficult to learn if the representational space is sparsely populated by examples, and the ability to form complex categories by combining many distinct representations to form one new category, all of whose elements are 'similar' is useful in the analysis and learning of structure in many domains.

Whether similarity should be viewed as a quality emerging from a complete theory, or prior to finding a theory of a domain, depends on whether one is interested in describing data using a particular theory, or learning a theory from scratch. If the latter is our interest, then observing similarity between representations can be a useful first step towards acquiring knowledge of the structure of the representational domain, which will help in the acquisition of a theory of the domain — a means of tackling the 'bootstrapping' problem. The problem of finding useful similarities empirically is now discussed.

2.7.1 Finding Similarity: An Example

Although I have said that finding similarity within a representational domain is very useful, since it allows us to infer a topology of the space, I have not yet given any examples of how similarity over formal spaces might be found using the insights of section 2.5. I shall now give an example of deriving a topology for a space by exploiting non-deterministic, denotational, and functional similarity.

Measures of similarity can be naturally defined for many representational systems. In a vector space, for instance, similarity of position is naturally captured by the notion of a 'metric', for which many candidates exist. In other systems, such as statistical observations of the output of some process, where no obvious metric is defined, statistical measures such as linear correlation, rank correlation, and the cross-entropy of the distributions of various statistics of the resultant representations can be used to give a measure of similarity between representations of items of interest. The similarities derived using this non-deterministic approach can be used, via the notion of *functional similarity*, to derive similarities for other, functionally dependent, spaces, provided that the regularity which exists between these spaces can be assumed to be continuous.

Figure 2.3 shows an item being simultaneously represented in two representational spaces, $\mathbf{R}_1$, and $\mathbf{R}_2$. As an example, we may take the state of a physical system comprising a weight on the end of a spring. Suppose that $\mathcal{I}_1$ maps such a physical system to a number corresponding to the length of the spring, while $\mathcal{I}_2$ maps the physical system to the mass in grammes of the weight, or even to a second, independent, measurement of the length of the spring (thus, $\mathcal{I}_2 = \mathcal{I}_1$. This case will be discussed below). We take a very large number of such physical systems, and represent them all in the way described above, we can draw a graph of $\mathbf{R}_1$ against $\mathbf{R}_2$. Assuming that the scales of the graph are linear with respect to both mass and length, by Hooke's law the resulting graph will comprise a large number of dots distributed around a straight line (the gradient of which depends on the strength of the spring). Now, it is possible to predict the mass of an item (element of $\mathbf{R}_2$) purely from knowing its representation in $\mathbf{R}_1$ (length by which it causes the spring to stretch). Indeed, this fact is taken advantage of by many kitchen scales. However, there is more that can be said about the relationship between $\mathbf{R}_1$

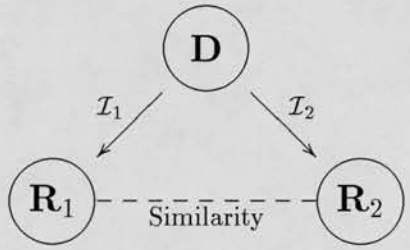


Figure 2.3: A special case of the general model of computation arrived at by imposing two intensional operators from the same denotational space, representing distinct but related aspects of that space. The two forms of representation, $\mathbf{R}_1$ and $\mathbf{R}_2$ will be informative about each other, and hence predictable from each other (to some extent). This informativeness may be exploited to impose a similarity metric between elements of both spaces. For an example relevant to this thesis, $\mathcal{I}_1$ might be the 'current word' of a newsgroup article, while $\mathcal{I}_2$ might be the 'previous word'. Alternatively, $\mathcal{I}_1$ might be the length of the extension of a spring, and $\mathcal{I}_2$ might be the weight on the end of the spring, or a second independent measurement of the length of the spring (see text).

and $\mathbf{R}_2$ than that they are just informative about each other. In fact, the relationship between the two representations is a continuous bijection, and can be used to impose a similarity metric over the space of weights ($\mathbf{R}_2$), given a similarity metric over the space of lengths ($\mathbf{R}_1$).

Suppose we can define a similarity measure over $\mathbf{R}_1$, but have not yet assigned one over $\mathbf{R_2}$ (suppose, for instance, we have a collection of rocks of unknown masses as our weights), then a naturally definable similarity measure over lengths in $\mathbf{R}_1$, $d_1(\mathcal{I}_1(x), \mathcal{I}_1(y))$, imposes a similarity between elements of $\mathbf{R}_2$ by defining $d_2(\mathcal{I}_2(x), \mathcal{I}_2(y)) = d_1(\mathcal{I}_1(x), \mathcal{I}_1(y))$. Supposing we wanted to present an analysis of how weight influenced extension, this definition of similarity has the advantage of respecting the *functional similarity* criterion discussed earlier.

If the number of possible representations in $\mathbf{R}$ is small, then it is possible to use another technique to simultaneously find similarity measures within the representational domains with respect to which the informational regularity between the two spaces is 'continuous'. This is an application of non-deterministic similarity due to the operator $\mathcal{I}$. A continuous regularity is one which maps similar items in $\mathbf{R}_1$ to similar items in $\mathbf{R}_2$. If we assume that the regularity which has been given to us is continuous in the true topology of the representational spaces, then we can use observations of this regularity

to impose constraints on the similarity metrics underlying the topologies of the two representational spaces. This can be achieved by exploiting some statistical properties of the nature of the relationship between $\mathcal{I}_1(x)$ and $\mathcal{I}_2(x)$.

In the example of the weighted spring, the regularity (due to Hooke's law) will give rise to a set of example devices which will, when extension is plotted against mass, give rise to a scatter diagram qualitatively similar to that in figure 2.4(a). The fact that it approximates a straight line is an artifact of the fact that the axes are represented so that similar weights appear spatially close to similar weights , and similar lengths appear close to similar lengths. (i.e. the topology of the representational space for weights and lengths is a total order, and so is topologically equivalent to a straight line). If the graph is redrawn so that the topological structure of the set of lengths and masses is no longer the same as that of the axis (e.g. the axes are remapped so that the order of weights and lengths on the axis is random), the graph is transformed to that of figure 2.4(b), which has no apparent structure.

However, if there are only 100 possible values of length and weight, then the original graph might become figure 2.5(a), and the random graph as in figure 2.5(b).[8] Let the set of possible values in $\mathbf{R}_1$ be $\{r_1, r_2, \ldots, r_{100}\}$. Each element of $\mathbf{R}_1$, r, will have been observed to be associated with some subset of values from $\mathbf{R}_2$. We call this subset of $\mathbf{R}_2$ the *image* of r, written $I(r)$. We define the *overlap* in $\mathbf{R}_2$ between $r \in \mathbf{R}_1$ and $r' \in \mathbf{R}_1$ as $|I(r) \cap I(r')|$ — that is the size of the intersection of the image of r and the image of r'.

The intuition is that if two elements have a high overlap, then they are similar, since they are associated with a similar set of elements of $\mathbf{R}_2$. Let us find an ordering on the elements of $\mathbf{R}_1$ which respects this similarity. That is we wish to find that linear ordering

[8]The reader may be concerned that quantisation is, in fact, cheating, since similar distances are mapped to the same "quantised" value, so some topological information is conveyed by the quantisation. It is true that this is precisely why the method works — the quantisation procedure does capture topological information. However, this is because real world systems tend to behave continuously with respect to the topological structure of the domain in which they operate. Indeed, this fact is why it is useful to find the topology of the domain. For instance, an analogue to digital converter is a device which quantises an electrical signal, and whose output we interpret as a number. The interpretation assumes a topology over quantised states (a particular linear order); the quantisation into discrete states respects whatever topology electrical voltage has (due to the physical nature of the device), but assumed no topology between these states.

which locally respects the quasi-metric $d(r,r') = \frac{I(r)+I(r')-2|I(r)\cap I(r')|}{I(r)+I(r')}$ (Assuming that $\forall r, I(r) > 0$, this expression is 0 if $r = r'$, and satisfies the other criteria necessary to be a metric, although it is quasi, since $d(r,r') = 0$ does not imply that $r = r'$). In the linear case, one way of ensuring this is to ensure that $\sum_{i=1}^{100} d(r_i, r_{i+1})$ is minimised, which corresponds to ensuring that the distance between adjacent items is relatively small (i.e. high overlap). Typically, the statistical metrics available for this technique allow only local structure to be uncovered — for instance, consider the metric, d, over lengths derived from the mapping between lengths and weights as described above. Although d(10,11) might be reliably less than d(10,13), d(10,20) will almost always be 1 in this metric (no overlap), which is the same as d(10,100). So only local structure can be reliably uncovered using such statistics.

Supposing the statistically defined metric for $\mathbf{R}_1$ reliably finds local structure, and some structure building system[9] finds a topological space which (approximately) respects the local structure uncovered by this metric, then there are two possible ways of finding a metric for $\mathbf{R}_2$. The first one is to utilise non-deterministic similarity again to find a metric for $\mathbf{R}_2$ by simply interchanging $\mathbf{R}_1$ with $\mathbf{R}_2$ in the process described above. The second one is to utilise the structure already uncovered, and the assumption of continuity (or, rather, isomorphism) of the regularity between the two domains in order to impose a topology on $\mathbf{R}_2$ by defining $d(f(s), f(s')) = \overline{d(s,s')}$, where $\overline{X}$ represents arithmetic mean of X, and $f(s)$ is a value of $\mathbf{R}_2$ associated with $s \in \mathbf{R}_1$.

A graph built in this manner looks like figure 2.5(a)[10], and so the order derived from applying this procedure will be very similar to the standard (numerical) ordering given to $\mathbf{R}_1$ and $\mathbf{R}_2$. A more detailed analysis of this is provided in section 6.2.1.

This procedure does not require any prior knowledge about the denotations of the representations in order to derive a similarity measure (such as that they are in corre-

[9]Many such systems exist in the statistical and neural-network literature. Three such methods are *multidimensional scaling* which maps elements into a vector space of a certain dimension so that the distances between the elements are preserved, *hierarchical cluster analysis* which, from similarity judgements, derives a hierarchical classification such that elements which are similar are likely to have a recent common ancestor, and Willshaw & Durbin's elastic networks, which map elements onto nodes of a neural network such that they maximise a measure of "conformity" between the empirically derived distances, and the topology of the network.

[10]Or one of three other graphs obtained by reversing the ordering of the axes, so that they run from high to low, rather than from low to high as is standard.

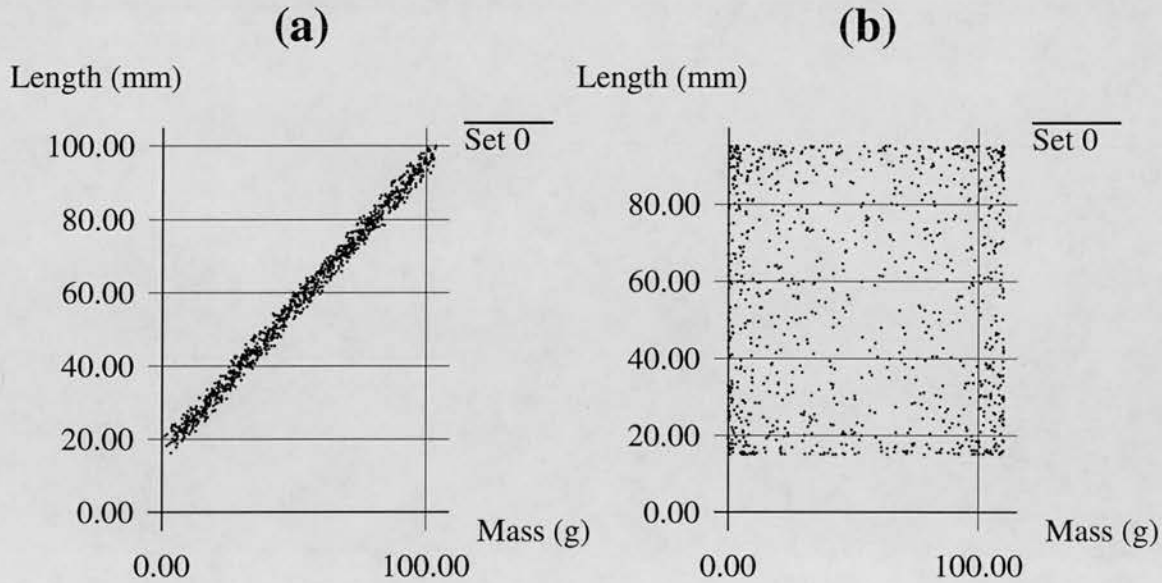


Figure 2.4:
(a): Graph of mass against length of a weighted spring such as that used in an experiment to verify Hooke's law. Measurement errors, and the fact that friction and other physical forces make Hooke's law only an approximation to reality account for the fact that the graph is not a straight line. The fact that it is clustered around a straight line is an artifact of the fact that the scales are ordered so that similar measurements appear close together.
(b): The right hand figure is from the same data as the left hand figure, but the point $\langle x, y \rangle$ is remapped to the point $\langle F(x), G(y) \rangle$ where F and G are functions which do not map similar numbers to similar numbers. This corresponds to having a different *intensional mapping* between lengths and numbers, and weights and numbers, than the standard one.

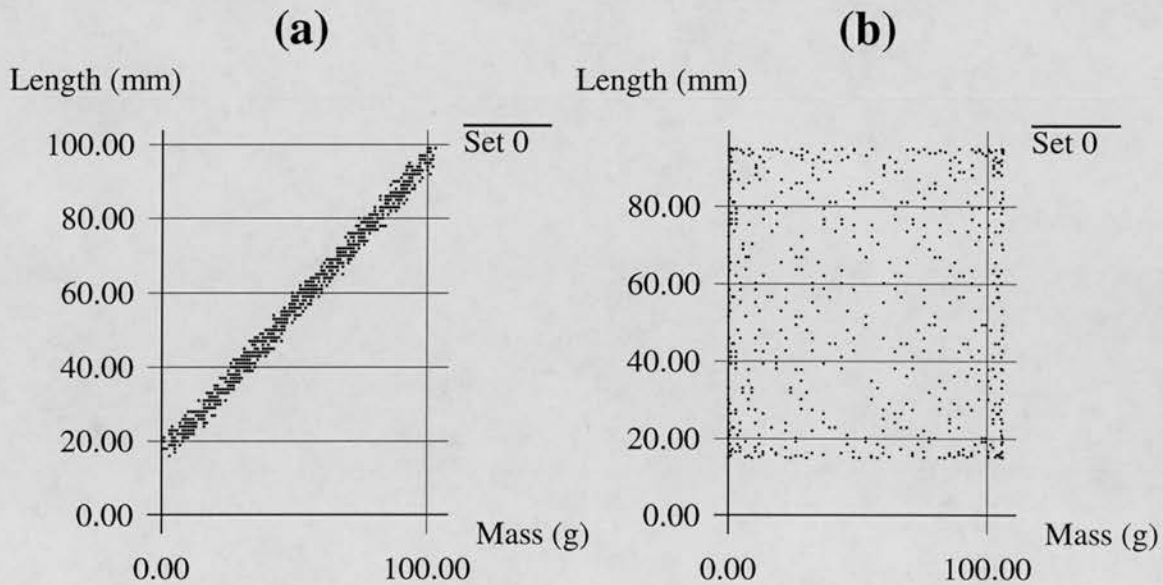


Figure 2.5: This graph is similar to the previous graph, but instead of allowing arbitrary values for mass and length, the scales are 'quantised' so that there are only 100 possible values for each. Thus the left hand graph displays a quantised version of Hooke's law, but it also has the statistical property of (roughly) maximising the overlap between adjacent values of mass, of the possible values of length. Thus, from a graph presented in random format (graph (b)), one can retrieve an approximation to graph (a) simply by finding the ordering of possible values of length and mass which maximise the amount of overlap between adjacent values. Thus functional similarity can be exploited to derive a topology for the representational spaces.

spondence with the real or natural numbers), and this method is very useful where it is not known a-priori what is being represented. All that is needed is a set of example pairs $\{\langle \mathcal{I}_1(x), \mathcal{I}_2(x)\rangle\}$, and some statistical method for determining how similar sets of values in $\mathbf{R}_2$ are to each other. This latter measure can be defined statistically, for instance by overlap (size of intersecting set) as in the previous example. Chapters 5, 6, & 7 concern themselves with the investigation of this point where language is what is being represented in various forms.

If $\mathcal{I}_2 = \mathcal{I}_1$, the observed set of independent identical measurements of the same system, the exhibited regularity is due to the non-deterministic nature of the intensional operator, $\mathcal{I}$, and the resulting topology is of a more fundamental nature than the *functional similarity* described above. It is clear that similarity due to such non-determinism constrains any functional similarity which can be detected (due to some other intensional operator, say $\mathcal{I}_3$), because all measurements are subject to the similarity due to this non-determinism. Thus, to take a famous example from chaos theory, if a pendulum is swung above two magnets, there are two stable states — the pendulum comes to rest over one of the magnets. Around some starting points of the swing of the pendulum, spatially close (and hence denotationally similar) locations give rise to a different stable state of the system (the pendulum stops over the other magnet). Thus it is not possible to usefully define areas of space corresponding to points which correspond to a particular stable state (functional similarity) because it is not possible to determine whether a particular magnet is in such a position because of denotational non-determinism (measurement error). Of course, a formal analysis can be made of an idealised system, but such a mapping can never be *learned* from example in this way.

All the components of this approach to discovering the structure of an a-priori unknown domain are well-known. Statistical measures of similarity between distributions of random variables have been proposed since the beginning of the subject, correlation coefficients being one such example. Also, structure deriving methods such as multidimensional scaling (Bechtel, 1976) and hierarchical classification (Sokal & Sneath, 1963) have been used widely in the social and biological sciences to derive structure from data. However, no one seems to have proposed the combination of these methods as a means

to learn the structure of a domain, whether as a proposal for learning in animals, or as a technique for learning in artificial intelligence, but rather the concentration has been in using these techniques to interpret data.

2.8 Review

The Marrian model of computation was analysed, with particular reference to the procedures used to find a representation of information in the outside world (intensional operator), and the nature of representational spaces. Computation can be viewed as something with external (denotational) relevance, as an abstract procedure or algorithm for manipulating representations of information, or as a physical system which implements some computation.

According to this view of computation, in order to perform computations within a domain it is necessary to represent information about it in a perspicuous manner. In many cases, this is facilitated by defining some topological structure for the representational domain. If, however, the topological structure of the domain is unknown, it is necessary to propose procedures which can find this structure empirically from a much simpler (and non perspicuous) representation scheme (such as an unstructured, discrete representation).

The local topological structure of such domains can sometimes be extracted by exploiting pertinent statistical regularities between simple representational domains, by utilising a statistical notion of similarity due to the nature of the randomness of the mapping between the simple domains. This is the approach which will be used to uncover linguistic structure later in this thesis.

Chapter 3

Language, Linguistics, and Statistics

Since the Chomskian revolution, it has become apparent that the structure of natural language is very complex. In order to explain the presence of the various structures found in natural language, and the absence of the ones which aren't found, theories have been proposed which posit complex hierarchical structures to represent words and sentences which are operated upon by various transformational rules (Chomsky, 1957), or recursively generated by the application of grammatical rules to lexical representations of the words in a sentence (Gazdar et al, 1985). While this approach has been (arguably) successful in describing many of the intuitions of language users, the richness of the posited structures and rules has been taken to have strong nativist consequences for language acquisition, whether by human or machine (Chomsky, 1965; Gold, 1967).

This chapter discusses the structure of language from a linguistic point of view, both from a structuralist and transformational perspective, and reviews some of the literature relevant to the machine learning of natural language.

3.1 Data

Some sequences of words can be used as sentences, and some sequences of words cannot be used as sentences. The sequence 'colourless green ideas sleep furiously', for example, is a perfectly reasonable sentence, while the sequence 'sleep green furiously ideas colourless' is not. *Grammar* is the subfield of linguistic study which deals with the explanation of how it is that some sequences of words are, and some sequences of words are not sentences, and for those sequences which are sentences, how to sub-analyse them into the combination of smaller components. As with all disciplines purporting to be scientific, it is necessary to define what data will be explained by the theories of the discipline.

The data which modern theories of grammar seeks to explain concerns the intuitions of competent language users regarding the acceptability or otherwise of sequences of words as sentences. The explanation is usually given in terms of linguistic categories and structures which are associated with words and phrases. For instance, if a linguist wanted to explain why

(3.1) This sentence is good.

is fine, but

(3.2) *This a string of words is good.

is not fine, they might appeal to the fact that a noun phrase ('this sentence') may only have one determiner, so 'this a string of words' cannot be a noun phrase, therefore (3.2) cannot be a sentence. Linguistic argumentation depends on being able to assign a valid syntactic structure to a putative sentence. If this is possible, the sequence can be interpreted as a sentence. If this is not possible, it can't.

3.1.1 Parts of Speech

The structures which linguists assign to sentences involve many linguistic entities. Words, at the lowest level, are classified into parts of speech, and thence into phrases

or groups, which are themselves assigned phrasal classes. Central to modern linguistic theories of the structure of the sentence is the validity of assigning words a class, and sequences of classes a structure as a phrase[1]. This will be more fully described in section 3.1.3 below, but there have been many attempts to justify linguistic notions of parts-of-speech, and the phrasal constituency of language.

From the paradigm of structural linguistics (e.g. Bloomfield, 1933), many tests have been proposed to define word classes. For the structural linguist, a standard series of tests was the means to define all the linguistic entities from the phoneme to the sentence, and find the classes associated with them. These tests relied on some principles of the structure of natural language, and although the paradigm of structural linguistics has been superseeded in linguistic theory, these principles remain. Prime among these principles is the *replacement test.* It states:

Definition 3.1.1 (Replacement Test (e.g. Radford, 1988)) *Does a word or phrase have the same distribution (i.e. can it be replaced by) a word or phrase of a known type? If so, then it is a word or phrase of that type.*

An example is the following: *red* and *pink* should be given the same category, since whenever it is *syntactically* legitimate to use the word *red* in sentence, it is legitimate to use the word *pink* in its place. However, *red* and *the colour of blood* are not syntactically equivalent strings, since although we can say:

(3.3) The red wine is good.

(3.4) The pink wine is good.

we cannot say:

(3.5) *The the colour of blood wine is good.

[1]It should not be assumed that all researchers who have followed this path have assumed that the classes to which words and phrases are eventually assigned should correspond to those which we have inherited from prescriptive grammarians (who in turn inherited them from Hellenistic Greek philosophers). See Harris (1951) for a fuller discussion.

Strictly speaking, the replacement rule only applies to linguistic *constituents*, and not to arbitrary substrings, and a number of other tests are necessary to (more or less successfully) determine constituency. So this, and other distributional tests, have led to a classification of words (and phrases) into a number of categories. These typically include **noun**, **adjective**, **article** (or **determiner**), **verb**, **preposition** and so on. Bloomfield and his followers, noted that these classes could be defined empirically in terms of whether the word could legitimately fill blank spaces in a set of skeletal sentences. Words which, when inserted in these skeletal sentences, gave rise to complete acceptable sentences could be put into an equivalence class together and called a single linguistic category, for instance, **adjectives** in the case of *red, green, happy* and so on.

One observation about the traditional linguistic classification of lexical items is that part-of-speech classification of a word is not determined by it alone. Often the part of speech a linguist would assign to a word varies according to its context of occurrence. For instance, the word *European* is assigned the category 'adjective' in

(3.6) I am a European native.

but a 'noun' in the phrase

(3.7) I am a native European.

The distributional evidence for this being with reference to other pairs of examples such as

(3.8) I am a British native.

(3.9) *I am a native British.

(3.10) I am a native Brit.

An even more clear example is the difference between the two occurrences of *her* in the following sentence:

(3.11) I gave her her boots.

especially when compared to to the difference between *his* and *him* in the following sentences:

(3.12) I gave him her boots.

(3.13) *I gave her him boots.

(3.14) I gave her his boots.

(3.15) *I gave his her boots.

(3.16) I gave him his boots.

(3.17) *I gave his him boots.

(3.18) *I gave his his boots.

(3.19) *I gave him him boots.

In fact, *her* can always occur wherever *his* or *him* can occur in sentences. Thus, the word *her* can be interpreted as a possessive personal pronoun (analogous to *his*) or as an accusative personal pronoun (analogous to *him*).

Parts of speech are often used to make generalisations about what other linguistic forms may be present in language. For instance in morphology, adding the suffix 'ness' to an adjective, x transforms the word into a noun describing the property of *being* x. Thus we have *redness*, *Europeanness*, *Britishness*, *Scottishness* but not *Britness*, *Scotness*, and so on.

So the map between lexical items and syntactic categories is non deterministic. However, for a set of about 30 general parts-of-speech tags, it is the case that 85-90% of word tokens in a large corpus are assigned their most common tag (Church 1992; Garside, Leech & Sampson 1987). Thus were the mapping between words and parts of speech to be assumed to be deterministic, the error rate in tag assignment could be made as low as 10-15%.

3.1.2 Closed Class Words

Of the linguistic parts of speech, some are 'open' in that new words may be added to them as language progresses, and some are 'closed' in that new words cannot be added to them. The class of prepositions, such as *of, on, in, ...*, is an example of a closed class. It is not possible to invent a new preposition. The class of nouns, such as *cat, dog, hat, paper, ...*, is an example of an open class, since new nouns are constantly being invented and incorporated into the language. It has been suggested (e.g. Powers, 1992) that closed class words serve as a 'framework' for the rest of language, and can be used as syntactic markers for the categories (and identities of) of content words.

It is interesting that most of the frequent words are, in fact, closed class words, and the frequencies of these are fairly stable across domains, while the relative frequency of open class words varies dramatically across domains. This implies that if we analyse the relation between words and the 150 most common words, we shall in fact mainly be analysing the relation between words and the closed class words, and according to Powers, this should be highly indicative of syntactic structure.

3.1.3 Theories of Grammar and Distribution

There have, through time, been many approaches to the study of grammar. To put the work presented in this thesis in the context of linguistic theory, I shall describe a meta-analysis of linguistic theories due to Halliday (1961) which unifies some of the comments on representation from chapter 1 with distributional linguistic theory and the statistical approach of this thesis.

Halliday suggests that linguistic descriptions explain phenomena on three *levels*. The level of *substance* is the actual material realisation of language, whether it be ink-blobs on paper, pixels on a television screen, or sound-waves in the air. The level of *form* is an abstraction over substance, and involves such linguistic concepts as the *word*, the *morpheme*, the *phrase*, the *discourse* and so on. The level of *context* concerns the relation of language to external events and conditions, for instance the situation being described by a sentence. There is a clear analogy to be found here with Marr's theory of

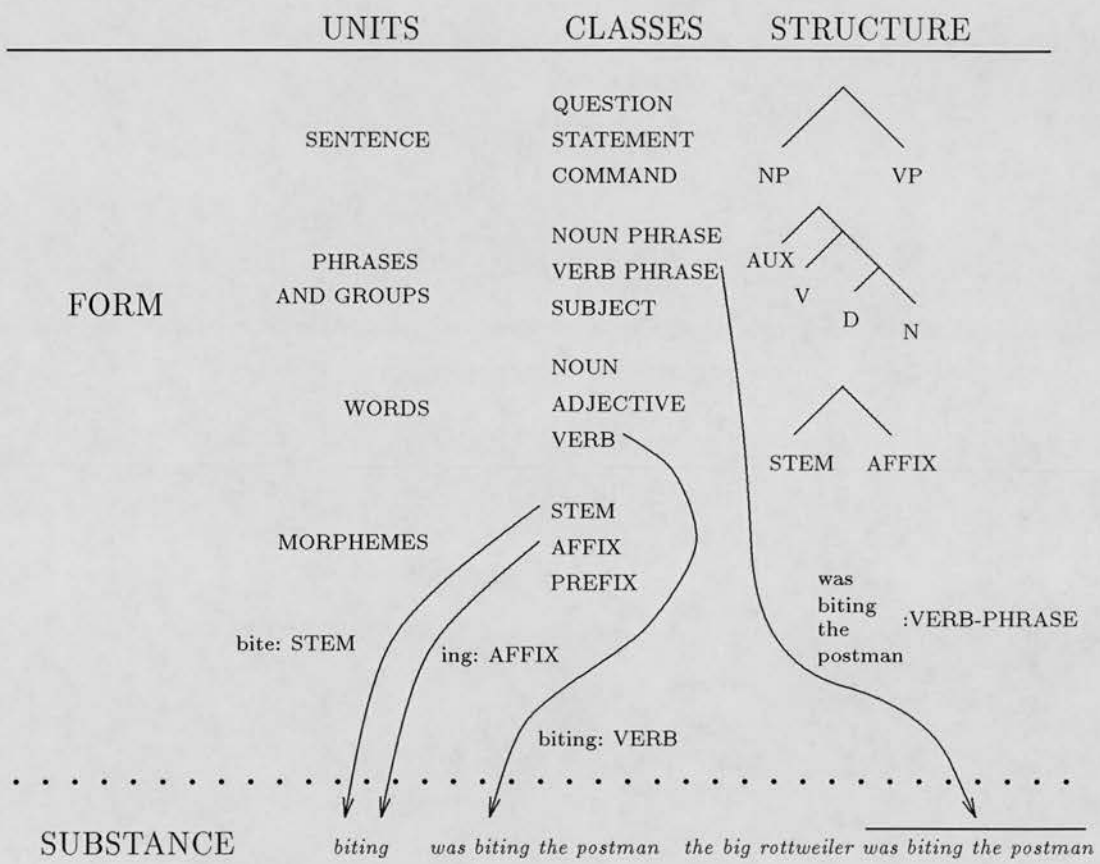


Figure 3.1: The relative position of the components in Hallidy's theory is illustrated in the case of a linguistic analysis of 'the big rottweiler is biting the postman'.

computation, which split computation into an external computational level (Hallidaean level of context), an internal algorithmic level (Hallidaean level of form), and a level of substance.

Grammar (and the theory presented here) operate at the 'system internal' level of *form*. The form of language has four inter-related components: The *unit*, the *category*, the *structure*, and the *system*. These will now be described in turn, and are illustrated in figure 3.1 for a linguistic analysis of *the big rottweiler is biting the postman*.

The unit: These are the fundamental *types* of items considered by the theory. Examples of linguistic units are the morpheme, the word, the phrase, the clause, and so on.

The class: Class is a means of subcategorising units. Examples include (from pho-

nology) *fricatives*, *glides*, *consonants*, *vowels*, *forward vowels*[2]; (from grammar) *nouns*, *adjectives*, *noun phrases*, and so on.

The structure: Structure is a means of describing complex linguistic units in terms of simpler ones. For instance, decomposing a phrase (e.g. *the big dog*) into a sub-clause analysis in terms of the categories of words (e.g. *(Det (Adj N))*). In simplest terms, the *form* of structures might be sequences, while for more complicated theories, structures might be trees, as above.

The system: Is the set of rules linking the unit, the class, and the structure together, and explaining one in terms of another. It is in the nature of systems that they are *closed*. That is, explanation of the structure of a sentence (and hence the rules of the grammar) depend only on reference to concepts involving the linguistic units, their classes, and the structures in which they participate.

For this thesis it is interesting to consider the relation between classes and structures. Structures are composed of classes of units, and the classes are determined by the role they play in the structures. Thus the attribution of 'noun' to *European* in example 3.7 above. Halliday writes:

> The relation between structure and class is a two-way relation, and there is no question of "discovering" one "before" the other. ... All structures presuppose classes, and all classes presuppose structures.

Since it is clearly part of the job of an acquisition system to give interpretations to both the classes and the structures of linguistic units, this is another example of the bootstrapping problem: Classes are licenced by the roles they play in structures, but structures are defined in terms of classes, and hence presuppose them. This thesis in no way rebuts this observation of Halliday's as a statement of linguistic theory. However, the statement from Halliday refers to a precise classification of a complete theory. It is unclear whether a partial or approximate classification of units can be achieved "before" a theory of the role these classes play in structures has been arrived at. Indeed, this thesis presents evidence that this, lesser, goal *can* be achieved.

[2]Forward vowels are vowels pronounced at the front of the oral cavity.

3.2 The Statistical Approach

Linguistic theory, as described above, deals with the *possibility* of strings of words being accepted as sentences of a grammar. An alternative approach, and the one adopted here, is to consider the analysis of sentences which actually do appear in large natural language corpora. One of the lessons of traditional linguistics is that the structure of language is complex; that sentences can usefully be decomposed into sub-phrases; that words can be assigned grammatical classes which determine the roles they may play in the structures of the sub-phrases to which they belong; and that these sub-phrases can be combined and transformed to form sentences.

One of the aims of this thesis is to show how a significant amount of this complicated linguistic structure can be recovered simply from the observation of very simple statistics of how users actually use language. The position adopted here is illustrated in Figure 3.2.

I adopt the assumption of standard linguistic theory that language use and understanding is based on complicated, hierarchical rules and representations of linguistic units (Chomsky, 1980), knowledge of the meaning of words, and extensive world knowledge. In the case of syntax, illustrated in the left hand side of Figure 3.2, it is usually assumed that language use involves a *mental lexicon* which stores the syntactic category of each lexical item (as well as semantic and phonological information), and a set of linguistic rules which specify the ways in which syntactic categories can be combined together. The relative weight attached to each of these components varies considerably between linguistic theories: according to traditional transformational grammar (Chomsky, 1957; Radford, 1988) and phrase structure grammar (Chomsky, 1957), the rules of syntax are of considerable theoretical significance; in some lexicalist accounts, by contrast, the syntax is extremely simple, and most of the explanatory burden is placed on the lexicon, as in categorial grammars (Lambek, 1958; Steedman, 1985) or lexical functional grammars (Bresnan, 1982).

The upper left hand side of Figure 3.2 denotes a fragment of a toy mental lexicon, and shows a syntactic analysis of a sentence made up from items in the lexicon, using

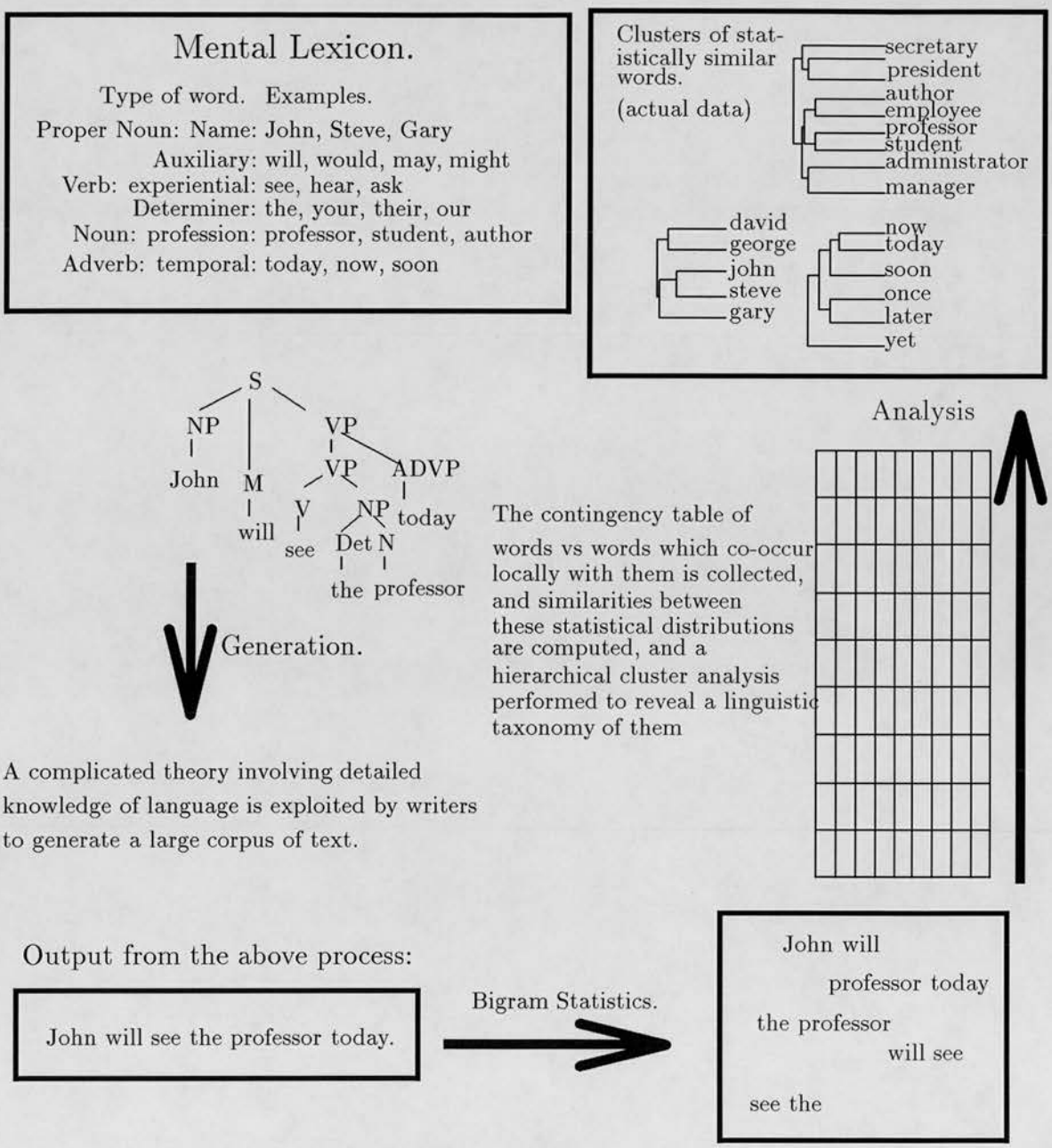


Figure 3.2: This figure outlines the paradigm of statistical analysis used in this thesis. The left hand side denotes a very complicated causal story underlying the production of natural language, but this gives rise to statistical 'footprints' which can be analysed using straightforward statistical techniques to uncover interesting facts about the causal process.

simple phrase structure rules (not shown). It is assumed that the derivation of some such syntactic analysis is involved in the generation of this sentence by the language user (following Chomsky, 1965), which is denoted at the bottom left hand side of the diagram.

Where the left hand side of the diagram denotes the *generation* of language using conventional rules and representations, the right hand side denotes the *analysis* of that language by statistical methods. The generated sentence reflects the underlying structure of the rules and representations that gave rise to it only very indirectly. The role of statistical methods is to help infer, from raw output data, aspects of the structure of the knowledge involved in language use. This is, of course, simply part of the general problem of language learning. The reason that *statistical* methods are appealing in this context is simply that the performance data is so untidy that a non-statistical rule based approach is liable to reject correct hypotheses concerning aspects of language structure, in the face of apparently contradictory data, which is actually simply caused by performance errors. The simple bigram statistics used are denoted in the bottom right hand corner of the diagram. Notice that all information about the overall order of the string of words has now been irretrievably lost; all that is known is which pairs of words occurred, and how often. This data is then incorporated into the bigram statistics of the language, and aspects of the structure of the mental lexicon (in this case, syntactic category) can be (partially) recovered by statistical analysis.

In the first instance, the system presented here is a method of statistical analysis which takes bigram information about the collocation of words in a large corpus, and converts this into a lexical hierarchy of words which, it will be shown, corresponds to an interesting linguistic taxonomy of them.

3.2.1 Stochastic models of the corpus

One approach to uncovering the structure of natural language is to assume that the natural language corpus was generated by one of a large number of priorly defined processes, and then use statistical inference to determine which one of these processes was most likely to have generated the observed corpus. If these priorly specified models have

linguistic interpretations, then this process of inference can be used to make linguistic inferences.

In terms of figure 3.2, the generating processes are possible instantiations of the left hand side of the diagram, and the process of statistical inference denoted on the right hand side of the diagram serves to infer the generating process given samples of its output. In order to use this direct statistical paradigm, one needs to define a class of candidate stochastic 'grammars', to find a means of estimating how likely a given grammar is to have generated a particular corpus, and to define a search procedure through the space of possible grammars which allows the best grammar to be found.

Clearly, finding a realistic model of the generating process is an onerous task, so a considerable amount of simplification must be made here. Typically, the corpus is assumed to have been generated by a Markov process, a Hidden Markov Process, or a Stochastic Context Free Grammar. For all these models, there exist (relatively) tractable procedures to iteratively find a locally optimal grammar from a corpus (Baum et al. 1970; Fujisaki et al. 1989).

The Shannon noisy channel model

The Shannon noisy channel model of dependency between random variables has been widely used since Shannon first proposed it (Shannon, 1948; Shannon & Weaver, 1949). It models the dependency between two random variables X, and Y by assuming there exists a "noisy channel" between them. A typical example of this is in sending digital information along a telephone wire, where there is a constant probability, p, of erroneously decoding a bit. This model of noisy communication is often used in conjunction with a model of what valid sequences are likely to be transmitted (i.e. what we expect to see) in order to calculate what sequence was most likely to have been transmitted given the data we receive. Brown et al. (1988) used this model to effect a language translation system, where an English text was passed through a Shannon noisy channel model to derive a French text. Brown et al. show that a remarkably linguistically naive system (in the sense of having a very simplistic model of English, and of the mapping between English words and French words) does very well on this task, out-performing many sy-

stems which have encoded considerable linguistic knowledge. However, a more relevant example of the Shannon noisy channel model for this thesis is the Hidden Markov Model (HMM).

A HMM is a model of language defined in terms of state transitions over a set of "hidden" states. Each state represents a class in natural language (for instance, nouns, verbs, etc.), and associated with the set of states is a simple noisy channel which maps states to observable words. The problem of inference for a system which tags words with their parts of speech is to take the observed sequence of words, and infer the hidden sequence of categories. This inference can be achieved by an algorithm due to Viterbi. Parameters can be fitted to this class of model from some training data either with or without initial knowledge of the true categories of words in the training data (i.e. tagged training data). Kupiec (1992) has shown that it is possible to train a HMM using the forward-backward algorithm from plain text provided only that it is known which *equivalence class* of parts-of-speech each word belongs to. Alternatively, if tagged training data is available, then the parameters on the HMM may be fitted by direct estimation (e.g. Jelinek 1986). In both cases, the reported accuracy of the resulting tagger is in excess of 96%.

The Markov property, which this model has, is remarkably simple. It assumes the probability of making a transition from one hidden state to another is dependent solely on the identity of the current state. This is linguistically naive, and is certainly not true of real natural language corpora, but nevertheless it shows the statistical importance of lexical identity and immediate context for the task of tagging natural language. The informativeness of the mapping between lexical items and the immediate contexts in which they occur, both in terms of parts of speech and individual lexical items, is the dependency which underlies the success of statistical systems in tagging natural language corpora.

3.2.2 Statistical Distributional Analysis

In traditional linguistics, the word "distribution" refers to linguistic intuitions as to whether a purported sentence is syntactically 'well-formed'. Well-formedness data is not

available (in general) to approaches based on analysing a corpus of uncertain integrity, but the replacement test can be made more empirically relevant by operationalising it as follows:

Definition 3.2.1 (Statistical Replacement Test) *Has the word or phrase been observed to occur in a corpus in similar contexts to another word or phrase? If so, then these should be given similar linguistic categories.*

This is an expression of a statistical regularity, or redundancy. It claims that knowledge of the current word or phrase should be informative about its context of occurrence. This is indeed the case, and it is this statistical redundancy which will be exploited to derive a classification of the lexicon. It remains to give formal accounts of what constitutes the "context" in which a word or phrase appears, and to define some measure of "similarity" between two such contexts.

The example of the spring in section 2.7 showed an example of how statistical regularity could be exploited to find topological structure in a representational domain with priorly unknown structure. There was a statistical regularity between representations of the weight on the end of the spring and representations of the length of the spring's extension, and this was exploited to derive a statistical measure of similarity which could be used to define a structure for the domains.

Now, we have the case that there is a statistical regularity (according to the Statistical Replacement Test) between words and the contexts in which they appear. Consequently one might expect a similar approach to work between representations of words, and representations of their context of occurrence. Consequently, the picture in the Marrian model of computation is as in figure 3.3.

Since observation of the *context of occurrence* is going to be the data upon which judgements of linguistic similarity are to be based, the definition of its representation is central, essentially determining what structure the resultant statistical clustering process can uncover. It is therefore desirable to have a representation of context which satisfies the following three criteria:

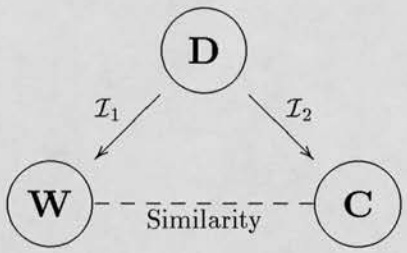


Figure 3.3: This figure shows how two representations are given for words which appear in a large corpus of natural language text. A word, together with its context of occurrence is what is to be represented, and hence is an element of the denotational domain, **D**. This is represented by $\mathcal{I}_1$ as a word in a punctate manner — each separate word maps onto a distinct element of the flat representational domain **W**. It is represented by $\mathcal{I}_2$ via its context of occurrence into the domain of contexts, **C**. This represents directly aspects of the context of occurrence of the word.

Availability The context should be readily computable from raw unlabelled data, and make as few assumptions about the underlying structure of the data as possible. If we make assumptions about the structure being modelled, then it should be no surprise if the eventual result of the computational system presented here confirms those assumptions. To take an extreme example, if the "context" was defined to be the orthodox syntactic label associated with the word in question, then a taxonomy could be found which would certainly respect linguistic orthodoxy, but this is only because linguistic orthodoxy was initially assumed, and is hence uninteresting. Moreover, if few assumptions about the underlying structure of the data are assumed, then the resulting learning paradigm is more likely to be applicable to other domains.

Expressiveness It is necessary that words of different syntactic category typically have contexts which are statistically differently distributed. Note that expressiveness does not refer to contexts — contexts are not the domain of representation we wish to find structure in; words are.

Linguistic utility Context should be mainly dependent on the syntactic category of the focal word. The 100th word before the focal word will probably not be informative about its syntactic category, but might rather be informative about its meaning (see Brown et al. (1990) for a discussion on this), so it is probably not

such a good idea to include such words so far away in the definition of context if we wish to uncover syntactic categories. However, the two previous and two succeeding words are informative of the syntactic category of the focal word, so representations of these in some form might prove very useful.

Statistical utility The statistical technique outlined in chapter 2 works best when the representational domain is small. This is because statistical measures of similarity either assume some structure over the representational domain a-priori (such as that it is a real vector space), or they make use of measures similar to the overlap statistic of section 2.7 which defines the similarity of two items according to the number of times they received the *same* representation. For a fixed number of items to be represented, having a small number of points in the representation space of contexts ensures that words frequently receive the same representation, and so increases the reliability of statistical metrics which might be used to find similarity.

Note that these criteria are very similar to the criteria presented in chapter 2 for candidates for representation. In fact, in a real sense, words will be represented by the observed distributions of their contexts.

The operational utility criterion has been split into two. First, there is linguistic utility which makes reference to linguistic entities. Since linguistic entities are what we want to derive, it is not possible to use sophisticated linguistic knowledge directly (otherwise I should be assuming what I wish to show), but general regularities such as 'closer words are more informative about syntactic category than words further away' can be exploited, so that although it is *statistical* regularity that is uncovered, since this statistical regularity is informative of syntactic category, the regularity can be interpreted linguistically. Secondly, there is statistical utility, which refers to the utility of the representation scheme for the purposes of finding non-deterministic similarity (see section 2.5).

Consequently, I used the four surrounding words as the definition of context, and further restricted attention to those surrounding words among the 150[3] most common words observed in the corpus. In line with the discussion earlier on closed class categories, the 150 most common words predominantly contain words in closed classes so it is likely that they will be particularly good indicators of the syntactic identity of the words around them (linguistic utility), and in line with the criterion of statistical utility above, using the most frequent words is likely to give rise to the most reliable statistics (statistical utility).

The context I use can therefore be thought of as four vectors of 150 dimensions, each dimension corresponding to one of the 150 most common words. The value of the vector is then the vector of the number of times the focal word appeared in the relevant relation (i.e. preceded, followed, was the last but one word, or the next but one word). Thus it is easy to compute, so it satisfies the availability criterion. The actual story is a little more complicated, in order to take the statistical replacement test defined above fully into account, but this will be detailed in the next chapter.

It remains to define empirical statistical similarity. It turned out that there were several candidates for this which were quite good at uncovering structure automatically. Candidates for the calculation of empirical similarity should also satisfy certain conditions.

Availability The definition of similarity should be easy to compute, and make as few assumptions about the nature of the underlying structure of the domain being modelled as possible. Again, it is no good assuming what is to be shown.

Linguistic similarity Empirical similarity should respect theoretical similarity. If a linguist would give two words a similar category, then the contender for the definition of empirical similarity should (usually) judge them as similar. Again, this is not possible to ensure a-priori, but statistics which are successful with respect to this criterion are to be preferred.

[3]This choice is entirely arbitrary. Although there are good reasons for using the most frequent word types, the choice of using only the 150 most common types was made for reasons of space-complexity. This is the first in a series of largely arbitrary design choices, and since the aim of the thesis is to show that unsupervised recovery of structure *can* be achieved, very little justification will be given for these choices.

Replacement criterion Empirical similarity should judge as similar words which are variants of the same word. Thus if the word "person" was randomly replaced with "pxrson" with a probability 0.5 every time it occurred in a corpus, and "pxrson" didn't otherwise occur in the corpus, then "person" and "pxrson" should be judged as similar, and different from other words.

The statistical content of this is that we require the definition of similarity to respect the statistical replacement test stated above. In the first instance, the formalism of this criterion will be couched in terms of a constraint on a distance measure, $d()$, between words in the lexicon which measures the linguistic similarity between two words. Words of the same class should be judged close, and words of different classes should not be judged close. The motivation for this interpretation will become apparent in later chapters. This might be formalised as follows:

Formal Replacement Criterion If every occurrence of a word, w is replaced throughout the whole corpus independently and at random by w' with probability p, and w'' with probability $1-p$, and neither w' nor w'' previously occurred in the corpus, then $d(w', w'')$ should be small.

This means, for instance, that the definition of similarity should be independent of the frequency of occurrence of individual words. This is not actually achieved in practice by the system presented here, because of the sensitivity of the definition of what counts as context to the actual frequency of words (the distribution of contexts being sensitive to only the most frequent words), and because expanding the number of word forms in a corpus means that a larger corpus is needed to get reliable statistics, but this is asymptotically true for the definitions of similarity used here as the size of the corpus gets large.

We also might insist that the definition of similarity should be insensitive to the addition of small amounts of "noise", or random sequences of words, to the corpus, since real corpora cannot, in general, be guaranteed to consist entirely of well-formed linguistic strings — there will typically be typing mistakes, grammatical errors, sequences of words which represent a vastly different use of the language to normal, and so on. This is not

a problem for classical linguistics, since such performance errors are ruled out of the domain of data to be explained by insisting that the data be from the *competence* of users' knowledge of language, rather than from their 'imperfect' linguistic *performance* as found in the corpora dealt with here.

Thus if a similarity metric d is derived empirically from some corpus C by some procedure F,[4] it is desirable that if C' is a noisy variant of C, that $F(C)$ and $F(C')$ should be similar to each other. In order to explore this idea, it is necessary to define a metric between corpora, and show that it produces a topology over corpora. In what follows, a corpus C, which is a partial mapping between the natural numbers and elements in some alphabet, $\mathcal{A}$, is said to have length n just in case $C(i) \neq \perp, 0 \leq i < n$, and $C(i) = \perp, i \geq n$.[5]

Definition 3.2.2 *Replacement corpus metric Let C and C' be corpora with alphabet $\mathcal{A}$ of length n_C and $n_{C'}$. The* replacement corpus metric $d_{\mathbf{C}}(C, C')$ *is defined as*

$$1 - \frac{\#_{0 \leq i < max(n_C, n_{C'})}(C(i) = C'(i))}{max(n_C, n_{C'})}$$

In all other cases, $d_{\mathbf{C}}(C, C') = 1$.

It is easy to show that $d_{\mathbf{C}}$ is a metric over the space of corpora with definable lengths.

We shall be interested in *metric inducing operators* which take a finite corpus and generate a metric between elements of a subset of the original alphabet of the corpus. In this regard, it would be fatal to the enterprise if the metric inducing operator were heavily dependent on being given input which is guaranteed to be syntactically "clean" (i.e. containing no grammatical errors, typographical errors, etc.), because such corpora do not exist. In fact, the metric inducing operator should be relatively insensitive to small amounts of "noise" in the corpus. The question is to what extent will adding small amounts of noise to a corpus effect the derived metric.

Let F be that function which takes a corpus and returns a distance metric between word types. Let $\mathbf{D}$ be the topological space of distance metrics between elements of the

[4] i.e. $d = F(C)$.

[5] This notation for corpora will be more fully detailed in chapter 5.

alphabet $\mathcal{A}$, which is some specified subset (not necessarily proper) of the alphabet of the corpora in the space $\mathbf{C}$. We define the *pointwise supremum metric* between distance metrics over $\mathcal{A}$ of the corpora $\mathbf{C}$ as follows:

$$d_{\mathbf{D}}(d_1, d_2) = \sup_{\psi,\phi \in \mathcal{A}} |d_1(\psi, \phi) - d_2(\psi, \phi)|$$

Again, it is easy to show that $d_{\mathbf{D}}$ is a metric. Now we are ready to define what we mean by a *noise resistant* metric inducing operator F.

Definition 3.2.3 *Noise Resistance*

Let $\mathbf{C}$ *be the space of all finite length corpora with alphabet* $\mathcal{A}_{\mathbf{C}}$, *with metric defined in 3.2.2. Let us consider some common subset of the alphabet* $\mathcal{A}_{\mathbf{C}}$, $\mathcal{A}$. *Let* $\mathbf{D}$ *be the space of metrics over* $\mathcal{A} \times \mathcal{A}$, *with the pointwise supremum metric.*

A metric inducing operator, $F : \mathbf{C} \to \mathbf{D}$, *is said to be* uniformly noise resistant to degree ρ within ϵ of C *just in case*

$$\sup_{C' : d_{\mathbf{C}}(C,C') < \epsilon} \{d_{\mathbf{D}}(F(C), F(C'))\} < \rho$$

This is a formal definition of the straightforward intuition that similar corpora should give rise to similar distance metrics. It is an overly strong definition in so much that even if a metric is not very uniformly noise resistant, it will be highly unlikely that randomly adding noise to C to derive C' so that $d_{\mathbf{C}}(C, C') = \varepsilon$ will change the induced metric by as much as is possible according to the uniform noise resistance criterion, which considers the *worst case* possible. For instance, if 1% of noise were added to a corpus of English, and this noise happened to change two thirds of the occurrences of "of" to some uncommon word, such as "doctor", then it is likely that the new distance between "doctor" and "of" will be very much smaller than before. However, this is a highly unlikely scenario, and can be fixed by changing the supremum operator to an expected value operator in definition 3.2.3, so that the definition considers the expected case of adding noise, rather than the worst possible case.

It turns out that all the measures proposed in this document are insensitive to relatively large amounts of such noise in this sense when inferring metrics over the two thousand most common words in a natural language, since they are based on statistical tests of inherently noisy data sources, and Zipf's law ensures that the 2000 most common word types are common enough to overcome small amounts of random noise.

3.3 Summary

The linguistic study of syntactic structure explains users' competence and the possibility of putative sentences, while the statistical investigation of structure operates over data from users' linguistic performance; the fact of what they have written or spoken. However, the classical analysis of language can inform the statistical investigation of linguistic structure insomuch that tests developed in the classical paradigm have natural statistical counterparts, and the general analysis of the syntactic structures which exist in natural languages suggests both stochastic models of the corpus which can be used in statistical investigation, and provides an analysis which can be used to direct a statistical attack on uncovering linguistic structure.

Essentially, classical linguistics defined what sorts of structures exist in language, and thereby defines the goal of a statistical learning system which seeks to uncover the details of these structures from the observation of redundancy in corpora. It should be noted, however, that some research groups define the goal of the uncovering of linguistic structure to be finding statistics which help *predict* what word comes next in the corpus, given knowledge of the corpus up til now. If this is the case, as it is with some groups working on speech recognition (e.g. Jelinek 1986), then the role of classical linguistics is lessened to one of suggesting statistical approaches to this problem, rather than defining what structure is to be uncovered. We shall be concerned with the former goal, and not directly with the problem of prediction.

Chapter 4

Language Learnability

Pinker (1979) observes that "how children learn to speak is one of the most important problems in the cognitive sciences". As a paradigm for research, the induction of rules for linguistic structure from tokens of speech and/or writing is relevant to all the main branches of the cognitive sciences: psychology (the process and fact of human language acquisition), linguistics (formal descriptions of what has been acquired), computational linguistics (especially the recently emerged branch of statistical computational linguistics), neural networks (both as inductive machines and as neurobiological models), Artificial Intelligence (techniques in automated machine learning). It is an interesting problem both in itself, and in the abstract as a complicated example of the problem of induction. Moreover, the recent ready availability of large machine readable corpora has facilitated the computer-based empirical analysis of language on a scale several orders of magnitudes above that previously possible.

First, a disclaimer: This thesis concentrates on the problem of machine learning of aspects of natural language, and accordingly how children learn language is peripheral. Nevertheless, that children can learn language, and learn it quickly, is an existence proof for its tractable learnability, and the way they learn it might inspire new approaches to the machine learning of natural language. No theory of child language acquisition, however, will be presented in this thesis.

This chapter asks what models of learning are the most promising for practical machine

learning systems. First, an analogy between learning a theory of language and scientific theories is set up. Then the literature of previous attempts, inspired by Gold, Chomsky, and others is discussed. The statistical approach to learning is discussed in this context, and used to motivate an approach to machine learning which takes statistical generalisation from data as its basis.

4.1 Scientific theory acquisition

The process of progress in a scientific discipline can be viewed as a process of theory acquisition and refinement. This is an illuminating analogy to draw with machine theory acquisition because much is known of the history of scientific enquiry, and the processes of scientific enquiry this history shows are at least available as the proper subject of research, rather than being in the domain of subjective introspection, as is largely the case with human theory acquisition, for instance. The analysis of scientific history presented here is roughly in accord with that of Popper (1959, 1970), Kuhn (1970a,b), and Lakatos (1970)[1].

Defining a scientific theory of a domain might be thought to involve at least defining what domain the theory offers explanations and predictions for, and what empirical evidence will falsify it. Experimental scientific enquiry, it might then be thought, should involve performing experiments which might decide between competing theories, while theoretical scientific enquiry should involve the proposal of new theories to replace the ones which have been experimentally falsified, and proposing theories which compete with the received theory and can be experimentally compared to it. This, however, is not an accurate empirical description of the process of scientific enquiry as it has been practised (Kuhn, 1970).

Rather, science can be better understood as a process of gradually acquiring larger and larger amounts of consolidated knowledge about the domain under study ("normal

[1] Although there are, of course, significant differences in views between these philosophers, all have a similar methodology — in order to find out what science is, look at how it has developed. Consequently, all agree on the fundamentals of the paradigm, namely that science consists of a number of periods of consolidation punctured by a number of 'paradigm shifts', where old assumptions are replaced by new ones. Further than this, they disagree.

science") punctuated at intervals by a number of "paradigm shifts" which radically change the nature of the theoretical foundation of the subject. During the process of "normal science", researchers do not seek to falsify the received theory in order to achieve scientific progress. Experimental researchers, rather, perform experiments which are expected to *confirm* the received theory, and can be interpreted within it (thus consolidating knowledge about it), or to collect facts about a domain for which there is currently no well-received theory, while theoretical scientists find new techniques to expand the domain of explanatory adequacy of an already received theory, rather than proposing competing alternatives. It is only at the small number of scientific revolutions (paradigm shifts) that the falsification model of science is an accurate description of the progress of scientific enquiry[2]. For instance, after Newton there was no fundamental change in the theory of mechanics for over two hundred years until Einstein proposed a competing theory. However, research in the field was concentrated on explaining various well-known effects *within* Newton's theoretical framework, and thereby expanding its domain of competence. This research was very valuable, explaining *prima facie* puzzling effects such as why water swirls a different way around a plug hole in the northern hemisphere than it does in the southern hemisphere. Nor, during this time, was there any empirical research aimed at falsifying Newton's theory. Rather, during this time, research was concentrated on discovering facts about electricity and magnetism for which there was no adequate theory and such empirical investigation of mechanics as there was, was concentrated on confirming the predictions of the theoretical scientists.

When paradigm shift did come in theoretical mechanics, Einstein said famously (and perhaps modestly) that it was due to his "investigating certain inconsistencies of Maxwell's (electro-magnetic) equations". However, in expounding his theory, Einstein made full use of the mathematical tools developed and consolidated in the 200 years since Newton first proposed the old paradigm. Therefore not only was the old paradigm instrumental in its own demise (by exhibiting inconsistency), but the tools it had generated through "normal science" were made full use of in the replacing theory. Indeed, it would seem difficult to imagine that Einstein could have proposed the theory

[2]Kuhn (1970) believes that even here, falsification is not an accurate description of scientific revolution. He believes that other factors (e.g. sociological, political) play a significant role in shaping scientific theory.

of relativity had it not been for the prior context of Newton's theory.

In this analysis, it is reasonable to regard a scientific paradigm (or theoretical framework) as consisting (at least) of a way of classifying objects in the domain under study, together with a set of rules defining the relationship between these entities. In providing an account of how such a theoretical framework might be arrived at in the first place, the history of science gives us few clues. This is because new theoretical frameworks are associated with paradigm shifts, and by their nature these are associated with the thought processes of individual scientists. The problem to be solved would appear to involve finding both new classifications of entities, and new rules defined over them. This involves solving a "bootstrapping" problem: the specification of a set of regularities presupposes a set of categories, but the validity of a set of categories can only be assessed in the light of the utility of the set of rules that they support. *Prima facie*, at least, this implies that both rules and categories must somehow be derived together.

One example of classification of a scientific domain which was the subject of much debate is the *periodic table* of chemical elements introduced by Mendeleev (1891), and refined by his followers. This classification of the chemical elements orders the elements by atomic number, and groups them according to similarity of chemical behaviour. This classification of the chemical elements underwent many changes in the nineteenth century.

The periodic table is an ordering of the atoms into a table of rows and columns. Across each row, trends are apparent. For instance the ratio of oxygen in the highest oxide of elements[3] in each row increases by $\frac{1}{2}$ as one traversed a row. The columns clump together elements whose chemical behaviour is similar. The question Mendeleev and his followers faced was "what is the best way to arrange the periodic table so as to maximise the number of chemical (and physical) regularities expressible *in simple (geometric) terms* within it?"

In the first place, the classification was based on atomic weight. Unfortunately, since atoms of the same element have varying numbers of neutrons, this classification is neither

[3]That is, the number of oxygen molecules which combine with each atom. For instance, if the highest oxide of iron is Fe_2O_3, then the ratio of oxygen is $\frac{3}{2}$.

well-defined, nor does it account well for many of the chemical trends which correspond to "laws" in this example. However, this simple classification allowed investigation to find trends which would lend support to a "better" classification; one more in accord with the known chemical regularities. In this way, refinement of Mendeleev's original concept was made, and eventually the periodic table as we know it was developed.

The story of the development of the periodic table, then, is an example of a drawn out mini-paradigm shift, which shows how a rough (though promising) initial idea can be refined by the analysis of experimental data into a more theoretically useful entity. It is instructive to note this as an example of how a "rough and ready" classification can be refined by experimental evidence to something which provided some of the main evidence (and inspiration) for later and complimentary theories: in particular in this case, models of the atom.

Normal science, then, proceeds by modelling more and more phenomena within the current paradigm, and hence explaining those phenomena. Paradigm shift happens when a competitor theory is developed which accounts for the data better than the received theory. One possible proposal for machine learning within a domain, therefore, might be inspired by the story of scientific progress: the machine sets up and refines a classification of the domain, and looks for regularities within the domain to expand its "knowledge" of it, making full use of skills it has already acquired by so doing. Periodically, it might radically change the theoretical basis of its knowledge, and this process might be due to discoveries made while it is expanding its knowledge in the old paradigm.

4.2 Formal-language acquisition

One proposal about how natural language might be learned by machine is that natural language should be treated as a formal language, and formal language learning techniques used. This proposal assumes that in order to learn language, one at least must be able to recognise and generate the strings of that language. This problem is equivalent to learning the grammar of that language, and hence we can concentrate on finding an

algorithm to determine a particular grammar from some large set of grammars solely from the evidence of example strings from the language (or more generally, from a teacher who will tell us whether or not a particular string is in the language). This section reviews the definition of formal languages, and the literature (mainly due to Gold) on how a machine might go about learning such a language.

4.2.1 The Chomsky hierarchy of formal languages

Chomsky (1957, 1965) considers the question of what classes of grammars are potentially able to account for people's judgements on the acceptability of natural language sentences. He proposes the so-called "Chomsky Hierarchy" of grammar formalisms — classes of grammar formalisms defined computationally which may or may not be powerful enough to account for people's judgement of acceptability of sentences of natural language.

Firstly, I define some terms for future reference.

Alphabet An *alphabet* is a base set of symbols which can be combined into sequences to form sentences of the language. In natural language, the alphabet might be the standard alphabet, or might be the set of all words in the language — the lexicon.

Language A language, $\mathcal{L}$, over an alphabet, $\mathcal{A}$, is a set of finite sequences of symbols from $\mathcal{A}$.

The set $\mathcal{A}^*$ is the set of all finite sequences of symbols in $\mathcal{A}$. Thus for all languages, $\mathcal{L} \subseteq \mathcal{A}^*$.

Grammar A grammar, $\mathcal{G}$, for a language, $\mathcal{L} \subseteq \mathcal{A}^*$, is a (computable) function which when given as input a sequence $s \in \mathcal{A}^*$, outputs 1 iff $s \in \mathcal{L}$, and 0 iff $s \notin \mathcal{L}$.

The Chomsky hierarchy of grammars is a hierarchy of computationally motivated restrictions on the properties of grammars. There are six levels in the hierarchy. These are now described from the most general to the least general.

Recursively enumerable functions Most general is the set of recursively enumerable functions. This class contains all Turing machines, and according to the Church-Turing hypothesis, all algorithms which might be defined. Unfortunately, this class of grammars includes elements which cannot possibly meet the specification of being a grammar for a language, since some functions, when given an input string, will never terminate — that is, they will never output either 1 or 0. Clearly, such functions cannot be grammars for any language (by definition), so the next level of the Chomsky hierarchy is precisely the class of grammars which meet the definition of being a possible grammar.

Decidable grammars This level consists of those functions which always make a decision about the acceptability of a sentence in a finite time: the *decidable grammars*. However, due to the *halting problem* there is no way of knowing, in general, whether an algorithm will always terminate (make a decision); no way of algorithmically determining (computing) this set. Since the strategy of grammar learning involves searching through a set of possible grammars, if this set is so large as the decidable grammars, and the space of possible grammars is itself computable, it must also include undecidable elements. Consequently, it is useful to be able to define subsets of decidable functions which are themselves computable.

Primitive Recursive grammars A further computational restriction is to the class of *primitive recursive* grammars. Although it is known that there are some interesting decidable functions not computable by a primitive recursive algorithm (for instance, the Ackerman function), it is to be hoped that acceptability decisions for strings in natural language fall into this category, since the rest of the Chomsky hierarchy is a subclass of this set. The important point about this class is that it is itself recursively enumerable — that is, there is a function which will output all elements of this set, and all the elements of this set satisfy the conditions of being a grammar for a language.

The remaining categories of the hierarchy are most easily obtained by more strictly defining the form of grammar formalisms[4], and imposing a series of ever stricter

[4] Although they do also correspond to computational restrictions.

constraints over these.

Context sensitive grammars The most general of these levels is the class of *context sensitive* grammars. Although I shall not describe them in detail, grammars in this class might be characterised as having a set of terminal symbols, a set of non-terminal symbols, and a set of expansion rules which detail how one non-terminal in a string might be transformed into another string. Context sensitive languages can have rules which refer to the (finite) context of occurrence of non-terminal symbols within a string, and which rewrite only one of these non terminal symbols. For example, a context sensitive rule might be written as

$$[\text{SINGULAR}]\text{VP} \rightarrow \text{V}s$$

meaning that in the prior context of a non-terminal SINGULAR, a non-terminal VP can be re-written as V followed by the terminal symbol s.

Context free grammars *Context free* grammars, on the other hand, can only refer to the identity of the "current non-terminal symbol" in a string in order to define rewrite rules. Thus, there could be no reference to the context of occurrence of the non-terminal which each rule rewrites, as there can in context-sensitive grammars. Context free grammars are powerful enough to define grammars for the artificial language $a^n b^n$, but not powerful enough to define grammars for $a^n b^n c^n$.

Finite state grammars *Finite state* grammars are defined to be decision procedures which can be implemented on a finite state automaton. They are equivalent to those context free grammars in which the rewrite rules are constrained to rewrite each non-terminal symbol with a string which contains at most one non-terminal symbol, and that must be at the end of the string.

Finite cardinality grammars *Finite cardinality* grammars are grammars of finite languages (and hence are equivalent to a simple list of valid sentences).

Associated with each grammar (except in the recursively enumerable class) is a language — the language which the grammar identifies. Consequently, we talk about *context free*

languages being precisely those identifiable by *context free grammars*, and so on for all classes of the Chomsky hierarchy. Gold asked the following question: What if all we know about a language is its position in the Chomsky hierarchy — computational restrictions on grammars which might identify it. How can this information be exploited to help us learn language?

4.2.2 Gold's learnability: Identification in the limit.

I will now present the *identification in the limit* paradigm for formal language learning, and comment on its applicability to natural language learning.

Gold (1967) considered the question of what it might mean to "learn" a grammar from example alone. He proposed that learning involved eventually (in finite time) finding a decision algorithm over sentences which always agreed with the grammar which generated them. That is, the algorithm which is found is an identification procedure which identifies all and only those strings licensed by the grammar. This approach he called "language identification in the limit".

In order to achieve this goal, the learner has access to the language, $\mathcal{L} \subseteq \mathcal{A}^*$, through one of two training paradigms. In the first one, the learner is simply presented with positive examples of sentences of the grammar. In this case, at time $t, t = 0, 1, 2, ...$ the teacher presents the learner with a sentence $s_t \in \mathcal{L}$. Every sentence in the language must eventually be presented by the teacher. In the second paradigm, the learner has access to language through a teacher who presents, at time $t, t = 0, 1, 2, ...$, a sentence, $s_t \in \mathcal{A}^*$, which either is or is not in the language, and tells the learner which is the case. Every sentence in $\mathcal{A}^*$ must eventually be presented by the teacher. Thus these paradigms differ insomuch that in the second case the learner has access to *negative evidence* about what is not in the language.

At all times, the learner must have a *current hypothesised grammar*, $\mathcal{G}_t$, which corresponds to learner's guess in the light of the data $\{s_i | i = 1, 2, ..., t\}$ about the unseen portion of the language, $\{s_i | i = t+1, t+2, \ldots\}$.

Whether it was possible to find such an algorithm, he found, depended on prior assump-

tions about where the grammar lay in the Chomsky hierarchy, and assumptions about which training paradigm was used.

The problem of finding the grammar which identifies a language has been reduced to one of searching through the relevant potion of the hierarchy, and the computational restrictions on grammar functions effectively restrict the search space.

4.2.3 Learnability in the Chomsky hierarchy

The question of where natural language lies in the hierarchy, and the possibility of learning a grammar for it solely on the basis of knowing where it lies in this hierarchy, is now discussed.

The simplest class of grammars are the finite cardinality grammars. These are grammars in which acceptable sentences are simply listed. Such grammars are not able to capture people's acceptability judgements for natural languages, since any such grammar must have a longest sentence (say of length n), and people can judge acceptable sentences of arbitrary length (such as "Bill thinks that Mary thinks that Peter thinks .... that Jo ate a sausage"), and in particular (at least) one sentence of length greater than n. One can, however, learn such a grammar from being immersed in a society of users of this grammar, simply by recording every sentence: eventually, in a finite (although a-priorily unknown) time, the entire grammar can be acquired in this way.

The broadest class of grammars correspond to recursively enumerable general rewrite systems. This class of grammars is very broad, and includes grammars for which there is no algorithm which can determine whether a given sentence could indeed have been generated from it. Although a grammar powerful enough to describe natural language must surely be in this class, it is impossible to learn all members of this class from example alone, since it is, for some members of this class, impossible even to decide whether a candidate string in generable from the putative grammar.

Between these two extremes are the five other classes. The second smallest set are grammars decidable by a finite state automata (FSA). Although all elements of this class can be learned provided the learning algorithm has the ability to ask an informant

whether or not a particular sentence is indeed licensed by the grammar, rather than merely being shown a set of sentences which are licensed by the grammar, this class does not contain grammars powerful enough to account for human judgements on certain "centre-embedded" constructions. The next three levels can also be learned in this way. They are the context-free grammars, the context sensitive grammars, and the primitive recursive grammars. It is unclear whether any (or all) of these contain grammars powerful enough to account for natural language. However, although it was thought for some time that language might be context free (Pullam & Gazdar, 1982), Shieber (1984) persuasively argued that, at least for some constructions in Swiss-German, no context free grammar would suffice. Chomsky had previously conjectured (1965, p. 61) that the species of transformational grammar necessary to describe natural language would not be found in the context-free class, but rather in at least the context-sensitive class, and no-one has disproved this. The final class of grammars, the decidable grammars, cannot be learned in the limit from example in either of the two training paradigms: to do so would be equivalent to enumerating the decidable grammars, making the class semi-decidable, which it is known not to be.

It is interesting to consider more closely how Gold proved these assertions. For the case of finite cardinality languages, the proof works because if all possible sentences are eventually encountered, a simple "look up table" may be compiled consisting of all valid sentences so far encountered. Eventually it is clear that the grammar can be acquired. In the case of the primitive recursive grammars (and all the subclasses of them in the Chomsky hierarchy), it is possible to order them, and find an algorithm which will generate the nth element of the class. Let us call the nth candidate grammar G_n. Now, either G_n makes the same judgements as the true grammar (in which case an agent having G_n as the candidate grammar will have learned the true grammar), or it makes a different judgement on a set of strings S. Since one of the strings in S (indeed, all of them) will eventually be presented by the teacher, a false G_n will always be falsified in finite time. The algorithm for learning by falsification in this way works as follows:

The language learning algorithm starts by hypothesising G_1, and looking at the input the teacher gives it. If ever the teacher gives it a string s inconsistent with G_1 (that

is, one for which G_1 make an incorrect judgement), it discards G_1 and chooses G_n, the least n consistent with the data gained from trials up to the current point[5], and repeats the process with G_n in place of G_1. This process continues for ever. It is easy to see that since any false grammar will always be rejected after a finite number of trials (since it will eventually always come across an inconsistent sentence), and that since finding the true grammar involves rejecting only a finite number of candidate grammars (since the true grammar is a candidate grammar of finite index), that this algorithm is guaranteed to find the true grammar in a (non-knowable) finite time. This is a possible schematic algorithm for the language learner:

The Gold Algorithm

```
START
   INDEX=0 /* Index used during searching */
   DATA={} /* The training examples seen so far */
   G=grammar(INDEX)
   REPEAT
      INPUT(sentence, judgement)
      DATA=DATA ∪{⟨sentence, judgement⟩}
      IF(G(sentence)!=judgement)
        WHILE(inconsistent(G, DATA))
           INDEX=INDEX+1
           G=grammar(INDEX)
END
```

Where **grammar()** is a function which generates a grammar given its index, and **inconsistent(.,.)** tests whether a hypothesised grammar gives correct acceptability judgements on a set of data items.

[5]The learner may do this by hypothesising G_2, and testing it against previous trials, and if that is falsified testing G_3, and so on until a grammar which is not falsified is found. Alternatively, it might have a more intelligent search method.

Learning and negative evidence

In order to learn a finite cardinality language using the Gold algorithm, no negative evidence is needed. This is because the learner never has to generalise — the conservative language which admits only those examples seen so far will suffice. In general, whenever the learner needs to find grammars which make positive predictions about an infinite number of sentences (or, in fact, if it ever generalises to unseen data), as is necessarily the case when learning non finite-cardinality languages, the learner needs negative evidence in order to determine whether the generalisation it has made includes positive evaluations of sentences which are, in fact, not in the language. The proof of this is simple. Suppose that after N positive examples, the learner guesses the correct grammar, $\mathcal{G}$ for the language $\mathcal{L}$. Choose a sentence $s \in \mathcal{L}$ which has not yet been presented as a positive example. Define $\mathcal{L}'$, with true grammar $\mathcal{G}'$, to be that language which is the same as $\mathcal{L}$ except it does not include s. According to the algorithm, since all future input will be consistent with the hypothesised grammar, $\mathcal{G}$, this grammar will never again be changed. Consequently, $\mathcal{G}$, and not the true grammar, $\mathcal{G}'$, is what is learned in the limit by the Gold algorithm. The Gold algorithm will never change its hypothesis unless its current hypothesis is shown to be false. Consequently any overgeneralisation it makes cannot be corrected by positive examples alone. Although this is not a proof that *any* learning algorithm needs negative evidence, Gold showed that any learning algorithm does in fact need negative evidence.

The proof that it is in general impossible to learn non finite-cardinality languages from positive examples alone is remarkably powerful, and the demonstration is in fact a proof that it is not possible to generalise in this paradigm. The proof relies on the existence of pathological presentation orders of positive examples for any non-finite language which ensures that a particular learning algorithm can't generalise. The proof runs roughly as follows:

Let $\mathcal{L}$ be *any* countable, non-finite language, and choose a chain of *finite* subsets of this language $\mathcal{L}_1, \mathcal{L}_2, \ldots$ such that $\forall n, \mathcal{L}_n \subset \mathcal{L}_{n+1}$ and $\bigcup \mathcal{L}_i = \mathcal{L}$. It can be shown that this can always be done if $\mathcal{L}$ is primitive recursive. By hypothesis, the learning function has to be powerful enough to learn in the limit every $\mathcal{L}_i$, as well as being able to learn $\mathcal{L}$,

from *any* countable presentation sequence of positive examples from the languages. It is not possible to achieve this, since for every i, presenting only a finite number of examples from $\mathcal{L}_i$ will cause the procedure to hypothesise $\mathcal{L}_i$ at least once by assumption (in fact, eventually always). So one chooses an order of presentation of examples from $\mathcal{L}$ so that the learner hypothesises $\mathcal{L}_1$ (ensuring that all examples from $\mathcal{L}_1$ have been presented at least once), then sometime later $\mathcal{L}_2$, then sometime later still $\mathcal{L}_3$, and so on. This can be achieved by presenting only examples from $\mathcal{L}_i$ until both $\mathcal{L}_i$ is hypothesised and all the members of $\mathcal{L}_i$ have been presented. Since one needs, by hypothesis, only *finite* amounts of examples to do each of these, there is a countable presentation sequence such that the learner will *never* eventually always hypothesise $\mathcal{L}$, since after any finite number of examples, n, the learner will later hypothesise $\mathcal{L}_n$, which is not $\mathcal{L}$. Moreover, it is clear that this is a valid presentation of positive examples from $\mathcal{L}$, since every $s \in \mathcal{L}$ appears in one of the finite languages, $\mathcal{L}_N$ (since $\bigcup \mathcal{L}_i = \mathcal{L}$), and hence is present in the presentation at least once.

It should, however, be noted that this result *does not hold* if assumptions are made about the order of presentation of examples. For example, if the examples are randomly chosen according to a certain constant probability distribution, the proof that valid generalisation is impossible fails. In fact, the proof works by continually changing the distribution of the presentation data (from examples from $\mathcal{L}_1$ to examples from $\mathcal{L}_2$, and so on). Such dependencies in the distribution of examples with time also interferes with statistical generalisation, so such a presentation would also stop statistical inference. It is clear that real life presentations are not so pathological, so the strength of the result is somewhat diminished.

4.3 Criticisms of Gold's paradigm

This paradigm of search presented by Gold is clearly not, as it stands, a practical proposal for the automated learning of a natural language. As Pinker (1979) points out, even considering grammars as simple as the class of finite state languages with seven terminal symbols (words) and seven states, there are more than 10^{100} possible

alternatives. Testing 1,000,000 of these every second, as the algorithm might do before moving on to more complicated grammars, would take far longer than the currently supposed existence of the universe. Pinker goes on to point out that

> The learner's predicament is reminiscent of Jorge Borges's "Librarians of Babel", who search a vast library containing books with all possible combinations of alphabetic characters for the book which clarifies the basic mysteries of humanity.

However, Gold has proved that no faster procedure for learning in the limit can, in general, be found. This is clear when one considers that after any finite sample, there will always be an infinite number of possible grammars not falsified by the sample. Consider two procedures following the Gold algorithm, differing only by the order in which they enumerate the grammars (i.e. the **grammar** function). If procedure 1 guesses the grammar after trial n, and procedure 2 after trial $m > n$, then consider another experiment with the true language being the language procedure 2 hypothesised at trial n. Now, procedure 1 will be incorrect at trial n, and procedure 2 correct. Consequently, for each of any two procedures we can find a language for which it finds the true language faster than its rival.

As a paradigm for learning natural language, it is possible to criticise Gold's approach on four counts: The classification of languages it uses; the criterion for success in learning; the susceptibility of the paradigm to "noisy" teaching data; and the general relevance of the paradigm to finding a practical machine learning system for natural language.

4.3.1 General Learning

It is scientifically desirable, at least from a psychological point of view, that the processes involved in learning natural language should be similar to the processes involved in learning about other domains. If systems which can detect interesting structure in, for instance, vision and sound, can be shown to be able to detect regularities in language, then these procedures might be assumed to be involved in learning about all three domains because this simplifies any general account of human learning. There is known to be very considerable local statistical redundancy in all these domains, and this is

evidenced by work in speech recognition (e.g. Huang et al, 1990), vision (e.g. Marr, 1982), and language (e.g. this thesis). Moreover, similar statistical models have been proved useful in uncovering structure in, and making predictions in, all these domains.

Opposed to this view is the 'nativist' view, which assumes that language has evolved as a highly specialised system, with its own rules and constraints, possibly unlike any other cognitive sub-system (Chomsky 1965; Fodor 1983). According to this view, a highly domain specific acquisition device might have evolved to acquire language quickly and accurately. This possibility is discussed further below.

From the machine learning point of view, it is far more interesting to suppose that a few learning strategies can be used to acquire interesting classifications and theories of many domains. From this point of view, even more than from the psychological point of view, it will be most interesting to view language learning as just one application of a set of general learning procedures which can acquire structure from appropriate descriptions of many domains.

4.3.2 Learning Natural language and the Chomsky hierarchy

Although the Chomsky hierarchy is a natural way to classify the learnability properties of formal languages, it is inappropriate when applied to natural languages. The main problem is that the Chomsky hierarchy is entirely insensitive to regularities which exist in finite cardinality languages, but classifies languages according to the form of the regularities which pertain over infinite sets of sentences.

To see this, consider the finite cardinality language, $\mathcal{L}'$, which consists of sentences in a non-finite primitive recursive language, $\mathcal{L}$, with fewer than N words. Let us also suppose that $\mathcal{L}$ is a natural language with considerable constituent structure. What do Gold's results tell us about the difference in learnability between $\mathcal{L}'$ and $\mathcal{L}$? They say that in order to learn $\mathcal{L}'$, no negative evidence is needed, but in order to learn $\mathcal{L}$, negative evidence is needed. Of course, if an individual were ever to learn $\mathcal{L}'$, their performance at recognising $\mathcal{L}$ would, in practice, be indistinguishable from someone who had learned $\mathcal{L}$, since $\mathcal{L}'$ includes everything that ever was, or ever will be written or spoken for, say

$N = 10^{(10^{100})}$.

The time needed to learn $\mathcal{L}'$, according to Gold's finite cardinality algorithm, which cannot be beaten for learning arbitrary finite cardinality languages, is proportional to $|\mathcal{L}'|$. If $\mathcal{L}$ is a natural language, it seems reasonable to conclude that $|\mathcal{L}'| \propto (\alpha|\mathcal{A}|)^N$, where $\mathcal{A}$ is the set of words of the natural language, and α is the mean proportion of words which can be substituted at a particular point in a sentence to produce another valid sentence[6]. Thus time taken to learn $\mathcal{L}'$ by a general algorithm, according to the formal language learning results must be exponential with N.

Now consider the following language, $\mathcal{L}''$. This language consists of $|\mathcal{L}'|$ random sentences from $\mathcal{A}^*$. Because any result from formal language learning theory refers to an entire class of the Chomsky hierarchy, it is clear that whatever the results from formal language learning theory say about learning $\mathcal{L}''$, they must say the same about $\mathcal{L}'$, which exhibits far more regularity than $\mathcal{L}''$. Given this, it is hardly surprising that the results from the formal language learning literature when applied to natural language *qua* formal language make for depressing reading. The regularities which are present in natural language are not adequately captured by the position of a natural language in the Chomsky hierarchy because the Chomsky hierarchy says nothing about regularities which occur within any given finite set of sentences, but only considers regularities within infinite sets. Although the two types of regularity are related, it is the regularities which exist within finite sets of sentences which must be exploited to learn language (even in Gold's algorithm), and the Chomsky hierarchy says nothing about these. To put it succinctly, although we know from formal learning theory that we can't learn *all* transformational languages, this is irrelevant because natural language is a particular transformational language. Moreover, what makes it special is the regularity which is evident over nearly all large *finite* subsets of sentences, and the Chomsky hierarchy does not classify these at all.

There have been several proposals for further constraining the set of candidate languages beyond the Chomsky hierarchy. One such is Chomsky's (1965) Language Acquisition

[6]This is closely associated with the "branching factor" of language, which is an interpretation of a measure called *perplexity*, which is a goodness measure for statistical language models. The value $\alpha|\mathcal{A}|$ is clearly lower bounded by 2 for $N > 4$, as can be seen from the sentence schema "my (father's | mother's)* cat is black.", which refers to a black cat owned by myself or some ancestor.

Device (henceforth, LAD). In this, the learner has a much reduced hypothesis set, corresponding to some "Universal Grammar". If the elements of this set can be enumerated, then learning can proceed as in Gold's paradigm. This approach has a strong nativist element (which should be avoided from the point of view of learning theory, if possible, because it is scientifically more interesting if the learning of language can be shown to be similar to learning of other things, since this produces a more unified theory of learning), and still requires an intelligent "parameter fitting" algorithm to find the correct grammar quickly. However, it may be that it is now not the case that an infinite number of grammars of this reduced set are consistent with any finite sample, and so, under certain conditions, language might be learned from texts of positive examples alone, rather than requiring the "negative evidence" needed by Gold's paradigm, and a learner might be able to know they have learned the language at the point where only one candidate grammar remains consistent with the examples. Work within this paradigm has recently been done (Fong & Berwick 1992) which sets parameters from a small set of examples to parse (and hence identify) strings from English and Japanese.

Alternatively, and more interestingly for this thesis, one might suppose that language has certain properties which make it easy to learn. Gold's paradigm involves the enumeration of possible candidate grammars. If it could be guaranteed that natural languages were chosen as candidates early in the procedure, reasonably assuming that the learning mechanism might be pre-specified (innate), then the pessimistic results of the formal language learning literature lose some of their force, and the theoretical generality of the learning procedure is not sacrificed. It may be, for instance, that language has evolved not an arbitrary grammar, but rather one which is easy to learn (is chosen early) by a moderately general learning procedure, characterised in Gold's terms by a particular enumeration of possible grammars. In order to ask "how is language learned", one might then ask "what special properties of language (as opposed to other sets of strings generated by grammars) ensure that its grammar can quickly be learned by such a general learning procedure?" We shall return to this point later, but it is surely connected to the great amount of regularity which exists in nearly all large, finite subsets of sentences.

4.3.3 Relaxing the learning criterion

Secondly, one might criticise the criterion for success in learning which Gold uses. It is not much more useful to be able to identify all sentences than 99.99% of sentences. In natural language in particular, it is very unclear that competent users of language can decide, for many strings of words, whether they actually are in the language or not. However, Gold's theorems still apply if one redefines "identification in the limit" by insisting that identification be made not of all sentences, but only of those which are definitely in the language, or definitely out of the language, and teach only those sentences. Depending on the definition of "definitely in/out", however, it is unclear where this re-defined natural language lies in the Chomsky hierarchy.

If one relaxes the criterion for success by accepting that a certain (predefined) proportion of sentences might be mis-recognised, another notion of language learning can be elucidated. We replace the "find an algorithm which accounts for all the data" goal with "find an algorithm which generates a grammar which is a close approximation to the real grammar". Wharton (1974) defines a metric over languages which makes this concept well-defined, and shows that a learning mechanism can succeed in learning a primitive recursive grammar in this sense using positive examples alone.

4.3.4 Integrity of the training data

All that has gone before supposes that the training data is free of noise. That is, the teacher always assigns the correct in/out judgement to each of the training sentences, or presents only sentences which are in the language. If this is not the case, Gold's paradigm falls down. To see this is the case, consider a teacher who misclassifies precisely one sentence, s. The grammar learnt in the limit must, according to the paradigm, also misclassify s. Thus Gold's learning mechanism learns what it is taught, and any failing in the teaching mechanism will cause the resultant grammar to be wrong.

Yet real texts (and speech) contain many sentences which are not acceptable sentences of language, and so even if Gold's approach could lead to the machine acquisition of language from pure error free teaching, it could not lead to the acquisition of langu-

age from real texts (although it might lead to the acquisition of some grammar which approximates the true language). It is a feature of real machine learning systems that we wish them to be as resilient as possible to "noisy input", and to learn the language in spite of a lack of integrity in the training data. One way to do this, as discussed above, is to restrict the class of candidate grammars more strongly, by defining a small set of possibilities a-priori, and finding the one *most* consistent with the training data. Another is by defining an a-priori probability distribution over grammars (for instance, according to their Solomonoff complexity), assuming a statistical model of the integrity of the training data which allows a non-zero probability of the teacher being in error, and choose that grammar which is most probable given the training data. Alternatively, there is the possibility of the combination of these approaches, both constraining the initial hypothesis space directly, and ordering grammars probabilistically according to some prior distribution. This is one of the approaches taken by statistical computational linguists discussed briefly in chapter 1.

4.3.5 Relevance of the paradigm to learning natural language

As a practical proposal for the learning of natural language, the formal language learning paradigm leaves a lot to be desired. Pinker's analogy with Borge's *Librarians of Babel* is telling — there are simply too many context free grammars for a blind enumeration method to work. Moreover, that the child learns language from sentences of no more than 1000 words long is an existence proof of a learning procedure which requires only a small fragment of the admissible sentences, and which operates largely without negative evidence.

There are several possible ways to change the assumptions underlying Gold's paradigm which result in more positive learnability results. Firstly, one can assume that the learner has an initial hypothesis space which is either finite, or vastly reduced with respect to the classes of the Chomsky hierarchy of formal languages. For instance, recent research in the framework of generative linguistics has concentrated on finding 'universal' linguistic descriptions which apply to all languages, languages differing with respect to the setting of a finite set of 'parameters'. If this is the case, then learning

language becomes simply a matter of finding a setting of these parameters which admits all the sentences heard so far, and when this is done, generalisation to all of language will be complete, and the grammar will have been learned. This strategy works because the initial hypothesis space of *possible* grammars has been vastly reduced to, in this case, a finite set. Consequently the search problem of finding the grammar has been made much easier. Gold's results no longer apply because one is no longer searching within the *entire* space of (for instance) context-free grammars, and so enumerations are finite. It may still be that negative evidence is required (for instance, if one setting of parameters generates a language which is a subset of language generated by another setting), but this would need proof, since Gold's proof no longer applies.

An alternative method is to use extra-linguistic knowledge to 'bootstrap' learning. It so happens that language is rarely used in isolation from what it is referring to, especially in 'motherese', the simplified language with which mothers speak to their children. Consequently, the relationship between what is spoken, and the learner's perception of the situation being referred to, can be used to generate *linguistic* hypotheses, even though these linguistic hypotheses might not make direct reference to that extra-linguistic information which was used to generate them. This is discussed more fully below.

Another approach to the problem might be to make use of the fact that there is considerable predictability of, for instance, the identity of the following word given knowledge of the identity of the current word. Statistical redundancy of this form can be exploited by well known algorithms to find the statistical analogues of context-free grammars from a corpus of positive examples of a language alone (e.g. Fujisaki et al. 1989). How successful these algorithms are in finding the grammar of a particular language depends on the nature of the statistical redundancies in the corpus, which are themselves largely dependent on the *complexity* of the grammar, measured in terms of the number of rules needed to define it.

If these statistical redundancies are expressed in complexity terms (after Solomonoff (1964) and Kolmorogov (1965)), and initial language *biases* are assumed which make some context free grammars much more likely than others, and make some example presentation schedules more likely than others, Adriaans (1992) has shown that it is

possible to find an algorithm which (probably almost) learns a context-free language from example in polynomial time. Although this is a major advance on Gold's results both in terms of a refinement of approach, and the utility of the result, I shall not discuss it further here because to do so would lead the discussion into the technical aspects of complexity theory which aren't central to the approach of this thesis.

It is interesting to consider again the analogy between science and learning. Gold's approach is similar to the "falsification" view of science, where alternative theories are generated, and experiments performed to falsify them until a theory is found which is unfalsifiable. Indeed, provided scientific theories could be assumed to be primitive recursive predictors of empirical data (which they may not be), this would be a possible paradigm for scientific research: simply find a grammar which generates precisely the empirically true statements within the remit of the discipline. However, there is no guarantee that any grammar so generated for physics, for instance, would resemble the scientific theory of physics in an interesting way. For any language, there may be many grammars which generate it. One might have an interestingly definable notion of "internal entity" corresponding to electrons, neutrons, energy, and so on, despite these terms not having appeared in the training language, while another might not. In terms of language, a Gold system might have learnt language, and yet not have an internal entity corresponding to "noun", "verb", "noun phrase", and so on — concepts so useful to our theories of language, and concepts by which, as we shall see, a large number of statistical redundancies in language are readily expressible. The converse of this latter observation is that these concepts are readily *learnable* by "fishing" for statistical redundancies in the natural language corpus.

4.3.6 Other work in the formal language learning literature

There has been much more recent work in formal language learning (see Wexler & Culicover, 1980 for a review). However, all the criticisms of the paradigm I have described apply to a great extent to all this work. The primary reason for this is the concentration on the Chomsky hierarchy as the means of classifying formal languages: the regularities which exist in natural language may well be expressible at some level of the Chomsky

hierarchy, but that does not mean that a language learner would need to learn (or even be able to represent) all languages at that level in the hierarchy. Indeed, it is clear that the only competent language learning mechanism we know about has great trouble learning even very simple finite cardinality languages (for instance, memorising lists), and since such may be "strapped on" to any language higher up the hierarchy, it follows that human beings cannot be thought of as general formal language learners.

Perhaps the most interesting extension of the formal language learning paradigm, from the point of view of applying it to natural language, has been made by Valiant (1984), who introduced the concept of *probably almost correct* (PAC) learning to the formal language learning paradigm. This replaces Gold's criterion of "identification in the limit". In order to PAC learn a grammar, an agent no longer has to provide a grammar which correctly identifies the entire language, but rather a grammar which, for any $1 > \epsilon, \delta > 0$, has to provide a grammar which, with degree of confidence at least $1 - \delta$, correctly identifies sentences a proportion at least $1 - \epsilon$ of the time. Thus the learning algorithm no longer has to be guaranteed to find a good identification algorithm, but rather just has to be able to find one with arbitrary degree of confidence. Moreover, the criterion for deciding whether an identification algorithm is acceptable as a grammar is not that the algorithm has to always agree with the grammar, but rather that the algorithm agrees with the grammar most of the time. If ϵ and δ are both made 0, then PAC learning reduces to identification in the limit.

If certain probabilistic constraints are imposed on the order of presentation of examples which, roughly interpreted, mean that simple sentences are more common than more complicated ones (which is empirically true), and that sentences are drawn at random from a population of sentences (which rules out the pathological sequences necessary to show that infinite languages cannot be learned from positive examples alone), then Adriaans (1992) has shown that context free grammars can be PAC learned in polynomial time with the number of rules in the grammar. Although it is not yet feasible to implement PAC learning procedures for natural languages, this is the most promising contribution of formal language learning theory to natural language learning theory. However, it is still problematical in the sense that the paradigm still requires an oracle

which says whether a sentence is in the language or not (i.e. negative examples), and to show that a language can be learned in polynomial time does not show that it is feasible to learn it in this way, since there are undoubtedly a very large number of grammatical rules in natural language, and no upper bound on the polynomial coefficients have yet been found.

4.4 Bootstrapping Learning

We turn our attention to regularities which might be able to help a language learning system to actually acquire knowledge of language. I have already argued that a Gold-type learning system is far too slow to be appropriate for real language learning. I have also suggested that there might be some special properties of language which make it more easily learnable by a general-purpose learning mechanism than an arbitrary context-sensitive language. This section discusses four approaches to this. Firstly, and briefly, the possibility that extra linguistic knowledge might make language more easily learned is discussed. Then the possibility that distributional regularities within language might be exploited to make language more learnable is discussed. Finally, there is a short description of prosodic and syntactic bootstrapping.

4.4.1 Semantic Bootstrapping

If the goal of a language learning system is not just to be able to identify languages, in the sense of Gold, but also to have a theory of how language encodes information (i.e. language meaning), then one might expect a language learning system to be able to make use of regularities between what is known about the situation being described, and the words used to describe this situation. If one supposes that information about the situation being described is available in some form of *mental representation*, then the task of learning language becomes the task of finding the mapping between sequences of words and the mental representations of this other, extra-sentential, information. Anderson (1976) considers the situation where the language learner has access to semantic representations of the situation being described, and details an algorithm

for acquiring language by exploiting the redundancy between linguistic descriptions of these situations, and their semantic representations. This procedure he calls "semantic bootstrapping", because semantic representations are used to induce a mapping between them and natural language sentences. Any well-formedness conditions over these semantic representations can then be used to induce well-formedness conditions over natural language sentences (for instance, "no crossing branches").

Although this is a good theory for how children might learn language, for the unsupervised recovering of linguistic structure, these semantic representations are not available, and consequently we shall not take this option. However, it is an intuitively appealing hypothesis, well supported by empirical data, which reminds us that the structure which should be assigned to natural language sentences (i.e. their representation) must be one which is computationally useful to other cognitive systems (e.g. systems which relate language to mental models of the world).

4.4.2 Distributional Bootstrapping

By contrast with semantic bootstrapping, *distributional bootstrapping* can be achieved solely with reference to a corpus of natural language. The first experiments in distributional bootstrapping were performed by Kiss (1972), who took a corpus of motherese which he collected by hand, and selected a small number of words for study. He then collected bigram statistics of the occurrence of these words for the relations 'previous word' and 'next word', and defined a metric between the statistics he collected over which he performed a cluster analysis (in much the same way as described in chapters 5 & 6). The resultant clustering showed significant differences between nouns, verbs, prepositions, and so on. He related the statistical significances of these clusters to the order in which children acquire words, and found that there was a high correlation between the order of acquisition and the order in which his algorithm clustered words.

The significance of Kiss's results should not be overstated. He showed that a distributional analysis could show differences between a small selection of words; he did not show that an entire classification of the lexicon could be achieved in this way, nor did he expand this work to handle short phrases. His aim was not to build a system capable of

the unsupervised acquisition of linguistic regularities, but rather to show how children might make use of such regularities to learn language. Also, he hasn't shown how a *grammar* might be acquired by similar methods. Very recent research by Redington (1992) has shown that the methods used by Kiss do generalise to give good results on word classification for large subsets of children's vocabulary. This approach to acquisition has also been proposed by other scholars (e.g. Maratsos 1982, 1990; Pinker 1984, 1987), sometimes under the name of 'correlational bootstrapping'. They generally reject it as a means of child language acquisition, because it does not take account of the semantic information available to the child. However, their arguments hold no force if the technique is viewed simply as a means to find some structure in language, and not as the sole component in a language learning system.

Recently, there has been work from Brill et al. (1990) concerning the automatic classification of the lexicon of the Brown corpus using similar (though not identical) techniques to that used in this thesis, and also work by Brown et al. in an internal IBM report (1990) along similar lines. Both show how distributional analysis can be used to derive approximate word classes from local distributional statistics of a large corpus.

4.4.3 Prosodic Bootstrapping

Information regarding the intonation and stress speakers assign to various words as they speak might be used to help assign structure to phonemic units, and hence to the structure of sentences (Morgan & Newport 1981; Kemler Nelson et al. 1989).

Although it is clear that prosodic information can be helpful in uncovering structure, and there is psycho-linguistic evidence to the effect that stress is correlated with syntactic structure, it is by no means clear that this information *alone* is sufficient to bootstrap the child's linguistic theory.

4.4.4 Syntactic Bootstrapping

This is Chomsky's (1965, 1986) innateness hypothesis that children are born with a 'universal grammar' whose parameters must be 'tuned' by the linguistic context the

child inhabits (usually parental speech). This has been discussed above, but whether or not the hypothesis is true, it is interesting to disregard it, and see how much structure can be extracted from natural language using one or more of the bootstrapping strategies listed above. If it is found that a grammar, or large parts of a grammar, can be acquired without resorting to this hypothesis, then the hypothesis will be unnecessary, or will be far less constraining on the space of possible grammars than is commonly assumed by those advocating this as the main method of bootstrapping language acquisition.

4.5 Discussion

We have seen that the task of learning an arbitrary context sensitive grammar is very difficult, and that there have been various proposals to exploit specific properties of language in order to make the task easier. The strategy I adopt is to take advantage of the fact that there is considerable local redundancy in *finite* natural language texts (i.e. words close to each other are highly predictive of each other), and this is a property of language which is *not* a property of an arbitrary context-sensitive grammar. The reason that prosodic and semantic information is not used is one of availability — large text corpora are available, while representations of their informational content, and of prosodic information are unavailable. The paradigm of learning I am interested in following is *unsupervised machine learning*, rather than an explicit search process through a specified set of hypotheses (as used by Gold), and in order to achieve this using statistical methods, large corpora are needed.

I drew an analogy between the process of scientific theory acquisition and the processes comprising an unsupervised approach to machine learning, and this is far removed from Gold's enumeration approach to learning, which as Pinker observed is reminiscent of the Librarians of Babel who search for the truth through the space of all texts. The paradigm of unsupervised learning, rather, might be thought of as a set of heuristic procedures which might be used to 'attack' the problem of finding structure within a set examples. To be sure, it will be hard to characterise those problems for which these procedures will be successful at uncovering structure, but it should be possible to

statistically detect whether significant structure has been uncovered or not, possibly by using what has been uncovered to build a model which predicts future examples with better than chance probability. If the prediction is much better than chance, then it is likely that significant structure has been uncovered.

The rest of this thesis describes some experiments aimed at uncovering some of the structure of language using unsupervised techniques. The extent to which it is successful will be evidenced by the quality, range, and robustness of the results gained from applying it. That is, its efficacy as a language learning methodology is an empirical, rather than a theoretical, matter.

Chapter 5

Theory of Statistical Classification

This chapter discusses the details of the theory of the experiments in unsupervised classification performed on corpora of natural language data. First, *streams* and *contingency tables* are formally introduced, and some operations defined over them. The general idea is that a contingency table of the current item in the corpus cross-classified with the item in some functional relation to the current item will be used as a gross representation of a corpus of natural language items (eg. words, letters, or phonemes) which can then be used for the purpose of classification. Observations of redundancy in the contingency table will be due to redundancy in the corpus, so classifications derived from the contingency table might be expected to provide useful classifications of the set of items under investigation. This, as shall be shown in the next two chapters, is indeed the case.

Then the chapter goes on to discuss the theory of analysing the contingency tables so as to uncover the structure inherent in them, exploiting the statistical regularities which pertain between two co-indexed representations of the corpus. Finally, the chapter discusses what statistics of the corpus might be expected to provide the best basis for syntactic classification.

The treatment is kept fairly general, and none of the theory presented here is specific

to natural language, being rather a general presentation of how statistical redundancy between mutually informative 'streams' can be exploited to uncover information about similarity between elements of those streams.

5.1 Streams, Contingency Tables, and Representation

Many computational experiments have been performed on corpora of various sizes, forming classification of various domains, on the basis of many contextual relationships, and with various similarity measures. First, we introduce some notation to allow us to describe the various stages of these experiments.

Recall from the discussion in chapter 2 that statistical regularities which exist between representations of different aspects of of a domain can be exploited to derive structure within a representational domain with previously unknown structure. Thus, we have a set of items, $X = \{x_1, x_2, \ldots, x_N\}$, and two (or more) intensional operators, $\mathcal{I}_1$ and $\mathcal{I}_2$ which represent different aspects of the items, and exploit the nature of the statistical redundancy between $\mathcal{I}_1(x)$ and $\mathcal{I}_2(x)$ for $x \in X$. In order to further analyse this position, we need to introduce notation for the set of items, X, the intensional operators, $\mathcal{I}$, and measures of the statistical relationship between the representational domains.

A corpus of natural language is a sequence of words, or possibly a sequence of trees which correspond to parses of natural language sentences. Such a sequence will be called a **stream**. The intensional operators, which are functions of the corpus, will be defined to be functions of the stream. The nature of the statistical redundancy between different intensional operators will be represented by a contingency table of the values of one representation verses the values of another representation.

I now define the term **stream**, and introduce some notation for the common functions of streams, so that new streams may be defined as functions of existing ones. The aim in doing this is not just to provide precise formal notation for the various procedures I shall later be describing, but also to show how the various techniques here might be generalised and still fall within the framework of the methodology described here. Examples of this general technique specific to natural language will be given in the next

three chapters.

Definition 5.1.1 (Stream) *A* stream, Ψ, *is a partial function from some indexing set,* $\mathcal{I}$, *to a set of symbols,* $\mathcal{A}$. *The set* $\mathcal{A}$ *is called the* alphabet *of* Ψ.

The symbol of Ψ *with index* n *is referred to by* $\Psi(n)$. *An element of a stream may be undefined, since* Ψ *is considered as a partial function between natural numbers and* $\mathcal{A}$. *The case that an element is undefined will be denoted by* $\perp$, *as in* $\Psi(0) = \perp$.

A finite stream *is a stream for which* $\Psi(n)$ *is defined for only finitely many* $n \in \mathcal{I}$.

Usually, the indexing set will be the natural numbers, $\mathbf{N}$, or the integers, $\mathbf{Z}$, in the case of a discrete stream of items such as natural language texts, but it is possible we might make use of a 'continuous' stream by making the indexing set the real numbers, $\mathbf{R}$[1], or even some other topological space (such as a list or a sequence of trees). For the purposes of this chapter, the indexing set shall always be considered the integers, $\mathbf{Z}$. Streams will therefore have many undefined elements, for instance $\Psi(x) = \perp$ for non positive x, if the element with index 0 is the first element of the corpus. The set $\mathcal{A}$ will be the set of word types in the language when considering similarity between words in chapter 6, but will be sequences or small parse trees when the theory is applied in chapter 7 to find phrasal structure, later in this chapter, streams will be defined where $\mathcal{A}$ has the mathematical structure of a multi-set, rather than just being a flat set. It is necessary to allow for undefined elements, because this simplifies the definition of streams derived from an original corpus. For instance, we shall later find it useful to derive streams from a corpus which are exactly the same as the corpus where $\Psi(i) \in \mathcal{A}' \subset \mathcal{A}$, but is undefined otherwise. Allowing undefined elements of the corpus considerably simplifies notation.

Thus we shall model a corpus of natural language data as a *stream*, Ψ, of words, letters, or phonemes. From one stream, it will be necessary to define others. For instance, we can define a new stream to be a pointwise function of an existing stream:

[1] For instance, if the corpus were a speech signal, rather than a sequence of words.

Definition 5.1.2 (Pointwise Projection) *Let $f : \mathcal{A} \to \mathcal{A}'$ be a partial function between the alphabets $\mathcal{A} \cup \{\perp\}$ and $\mathcal{A}'$. such that $f(\perp) = \perp$.*

The pointwise projection operator, $\mathbf{P}_f$, *is a function between streams, defined such that*

$$f(\Psi(n)) = \mathbf{P}_f(\Psi)(n) \tag{5.1}$$

That is, the nth element of the stream $\mathbf{P}_f(\Psi)$ is the projection under f of the nth element of Ψ.

This generates a new stream of characters, but there is a functional dependency between $\Psi(i)$ and $\mathbf{P}_f(\Psi)(i)$. It will prove very useful to define some functions, $\mathbf{F}$, over streams where the dependency between $\Psi(i)$ and $\mathbf{F}(\Psi)(i)$, although a function of Ψ, represents a sequential relation such as 'previous word'. A class of such functions are now defined:

Definition 5.1.3 (Displacement Operator) *We define the* displacement operator, $\mathbf{\Delta}_m$ *to be a function between streams with the same alphabet such that*

$$\mathbf{\Delta}_m(\Psi)(n) = \Psi(n+m) \tag{5.2}$$

Thus $\mathbf{\Delta}_m$ can be thought of 'shifting' the source by m places

It is clear that sometimes a displaced source will be undefined for indexes for which the original source is defined. For instance, if $\Psi(n)$ is defined only for non-negative integral n, then we would want

$$\mathbf{\Delta}_{-1}(\Psi)(0) = \perp$$

In order to classify lexical items from streams (which will be formal representations of corpora), we shall be interested in the dependencies between two or more streams of data. Therefore, it will prove useful to be able to talk about two or more dependent streams as a unit. This is simply the product operation over streams, which is naturally defined by:

Definition 5.1.4 (Product Operator, Π) *Given a set of streams,* $\{\Psi_i | i = 1, 2, ..., N\}$, *with alphabets* $\{\mathcal{A}_{\Psi_i} | i = 1, 2, ..., N\}$, *we define their product to be a stream with alphabet* $\prod_{i=1}^{N} \mathcal{A}_{\Psi_i}$ *as follows:*

$$(5.3) \quad \left(\prod_{i=1}^{N} \Psi_i\right)(n) = \langle \Psi_1(n), \Psi_2(n), ..., \Psi_N(n) \rangle$$

Sometimes we write $\prod_{i=1}^{N} \Psi_i$ *as* $\langle \Psi_1, \Psi_2, ..., \Psi_N \rangle$.

Thus the product of streams is a stream of products. Once we have such a product stream, we can note the joint distribution of occurrence of the values of the individual streams which make up the product stream. This is done by counting the number of times the values of the product steam occur, and putting them in a n-dimensional contingency table, where n is the number of streams forming the product stream.

Definition 5.1.5 (Contingency Table) *Let* $\Psi = \langle \Psi_1, \Psi_2, ..., \Psi_n \rangle$ *be a product stream (of possibly one stream). The* contingency table *of the product stream* Ψ, *is a function from n-tuples of alphabetic characters to natural numbers such that*

$$[\Psi](\langle s_1, s_2, ..., s_n \rangle) = X$$

where X *is the number of times that* $\Psi(m) = \langle s_1, s_2, ..., s_n \rangle$ *for* $m \in \mathcal{I}$, *where* $\mathcal{I}$ *is the indexing set of* Ψ.

Normally, we drop the angle brackets around a product source in a contingency table for succinctness, thus we write $[\Psi_1, \Psi_2, ..., \Psi_n]$ *for* $[\langle \Psi_1, \Psi_2, ..., \Psi_n \rangle]$.

Often, the contingency will cross classify a stream of items under study, Ψ, with a stream of alternative representations of those items, Φ. In this case, it will prove convenient to describe the items under study as the *focal* items, these corresponding to the alphabet of Ψ, and the associated stream to be the *focal stream*, while the other stream will be *peripheral stream*, and its alphabet the *peripheral* items.

By convention, the focal stream will be listed first. Thus when we write $[\Psi, \Phi]$, Ψ will be the *focal* stream, and Φ will be the *peripheral* stream.

It will also be useful to define streams which contain multi-sets from an alphabet. Just as sets can be (naively) thought of as functions from a universe of entities to $\{0, 1\}$, with $s(x) = 1 \Leftrightarrow x \in s$, multisets can be thought of as functions from a universe of entities to the natural numbers, with $s(x)$ meaning the number of times that x is in s, rather than just whether it is in s or not. Analogously to the concept of the power set of a set X being the set of all functions $X \rightarrow \{0, 1\}$, the multi-power-set of an alphabet $\mathcal{A}$ is the set of all functions from $\mathcal{A}$ to the natural numbers. Finite union can be defined as the pointwise maximum of multi-set functions, and general intersection as the pointwise minimum. This is a useful concept if, for instance, we wish to construct a representation of an article as an multi-set of its words, and define the corpus to be a stream of articles, or if we want to define the peripheral operator to be the set or multi-set of the words which appear within 1000 words of it. This idea will be useful in uncovering phrasal structure in chapter 7, and in uncovering semantic structure in chaper 6.

Definition 5.1.6 (Multi-Stream) *Given an alphabet, $\mathcal{A}$, we define the* multiset *alphabet to be $\mathcal{P}_M(\mathcal{A})$, which is the set of all multi-sets of $\mathcal{A}$. A stream Φ^M is called a* multi-stream over $\mathcal{A}$ *if its alphabet is the multi-power-set of $\mathcal{A}$. Its* alphabet *is $\mathcal{P}_M(\mathcal{A})$, and its* base alphabet *is $\mathcal{A}$. Such streams will always be superscripted with M to distinguish them from other streams.*

We also need a way to "cast" ordinary streams to multi-streams. This is easily achieved by:

Definition 5.1.7 (Multi-Stream Casting Operator, $\mathcal{M}$) *If Ψ is a stream (possibly a multi-stream) with alphabet $\mathcal{A}$, $\mathcal{M}(\Psi)$ is a multi-stream defined over $\mathcal{A}$ such that*

$$(\mathcal{M}(\Psi))(i) = \{\Psi(i)\}$$

$$(\mathcal{M}(\Psi))(i) = \{\} \text{ if } \Psi(i) = \perp$$

for all i in the indexing set of Ψ. The indexing set of $\mathcal{M}(\Psi)$ is identical to that of Ψ.

The definition of product streams remains unaffected, it being possible to form the product of streams with multi-streams straightforwardly. The pointwise projection operator can also be left as defined, although usually we shall wish to extend it so that a mapping from $\mathcal{A}$ to $\mathcal{A}'$ derives a mapping from $\mathcal{P}_M(\mathcal{A})$ to $\mathcal{P}_M(\mathcal{A}')$ in the natural way.

The only reason why multi-streams are explicitly typed is that the contingency table operation has a different meaning for them. If Ψ^M is a multi-stream over $\mathcal{A}$, the contingency table is defined as a multi-set over $\mathcal{A}$, *not* as a multi-set over $\mathcal{P}_M(\mathcal{A})$ as would be the case if definition 5.1.5 were applied to Ψ^M as an ordinary stream. Consequently, we refine the definition of contingency tables to handle the case of multi-streams. First we note than since a normal stream can be trivially cast into a multi-stream by defining $\Psi^M(n) = \{\Psi(n)\}$, we can define the contingency table as operating only over multi-streams. Whenever a contingency table is taken of a product involving both ordinary streams and multi-streams, we assume this casting operation to be implied.

Definition 5.1.8 (Multi-stream Contingency Table) *Let* $\Psi = \langle \Psi_1^M, \Psi_2^M, ..., \Psi_n^M \rangle$ *be a product of multi-streams. The* contingency table *of the product stream* Ψ*, is a multi-set over* $\prod \mathcal{A}_i$*, where* $\mathcal{A}_\rangle$ *is the base alphabet of* Ψ_i^M*. It is defined as:*

$$[\Psi](\langle s_1, s_2, ..., s_n \rangle) = \sum_{\lambda \in \mathcal{I}} \prod_{i=1}^{n} \Psi_i^M(\lambda)(s_i)$$

Normally, we drop the angle brackets around a product source in a contingency table for succinctness, thus we write $[\Psi_1^M, \Psi_2^M, ..., \Psi_n^M]$ *for* $[\langle \Psi_1^M, \Psi_2^M, ..., \Psi_n^M \rangle]$.

It is easy to show that (5.1.8) is the same as (5.1.5) when the $\{\Psi_i^M\}$ are trivially cast versions of ordinary streams, since $\prod_{i=1}^{n} (\mathcal{M}(\Psi_i))(\lambda)(s_i)$ is 1 just in case $\Psi(\lambda) = \langle s_1, s_2, \ldots, s_n \rangle$, otherwise it is 0.

For a stream, Ψ, the $\Psi(i)$ refers to the *current* item, $\Delta_{-1}(\Psi)(i)$ to the item *before* the current item, and so on. We now make two further definitions. First, we define the *restriction* operator, R_W, which is a special case of the pointwise projection operator.

Definition 5.1.9 (Restriction Operator, $R_W(\Psi)$ or $\Psi_{|W}$) *Let* $\mathbf{I}_W$ *be defined for a set* W *thus:*

$$\mathbf{I}_W(x) = \begin{cases} x & \text{if } x \in W \\ \perp & \text{if } x \notin W \end{cases}$$

The Restriction Operator, R_W, *is defined to be the pointwise projection of* $\mathbf{I}_W$, $\mathbf{P}_{\mathbf{I}_W}$ *(see definition 5.1.2). Thus* $R_W(\Psi)(i)$ *is* $\Psi(i)$ *if* $\Psi(i) \in W$, *otherwise* $R_W(\Psi)(i) = \perp$. *Sometimes it will be more convenient to write* $\Psi_{|W}$ *for* $R_W(\Psi)$.

For multi-streams, the restriction operator is defined over the base alphabet, not the power set.

The *restriction operator*, R_W, is typically used where measurements are only taken of (peripheral) contexts involving, say, the 150 most common items, or for limiting the number of focal items being considered.

It is useful to be able to talk about the values of individual cells in contingency tables, and for talking about marginals obtained by summing the values of the contingency table over a set of dimensions.

Definition 5.1.10 (Contingency Table Notation) *Write* $X = [\Psi, \Phi]$. *This, as defined in 5.1.8, is a contingency table. We write* $X_{\lambda\mu}$, *or* $[\Psi, \Phi]_{\lambda,\mu}$ *for the value of the cell of* X *which corresponds to the value of* Ψ *being* λ *while* Φ *is* μ *(formerly defined in definition 5.1.8 as* $[\Psi, \Phi](\langle \lambda, \mu \rangle)$

We write $X_{\lambda+}$ *for*

$$\sum_{\mu \in M} X_{\lambda\mu}$$

where M *is the alphabet of the stream* Φ. *Similarly we define* $X_{+\mu}$, X_{++}. *This notation extends to multi-dimensional contingency table in the obvious way.*

5.1.1 Common Operations on Contingency Tables

It will prove very useful to be able to change the values of cells of contingency tables once they have been collected by performing arithmetic operations over them, and to be able to perform structural operations on contingency tables, for instance being able to define new contingency tables in terms of old ones.

Arithmetic operations

These will be defined by giving an arithmetic expression for calculating the value of cells of a new contingency table given an old one. For instance, if the new contingency table, Y, is arrived at from a table X by enforcing the constraint that all the rows sum to 1, while retaining the proportions of values in the row, then a suitable operator, Ω, might be

$$\Omega : \Omega(X)_{\lambda\mu} = \frac{X_{\lambda\mu}}{X_{\lambda+}}$$

$$Y = \Omega(X)$$

A common 'normalisation' operator, Ω_E, will be defined as

$$\Omega_E(X)_{\lambda\mu} = \frac{X_{\lambda\mu}X_{++}}{X_{\lambda+}X_{+\mu}} \tag{5.4}$$

The value of each cell of $\Omega_E([\Phi, \Psi])_{ij} = \Omega_E([X_{ij}])_{ij}$ is $\frac{X_{ij}}{E(X_{ij})}$, where $E(X_{ij})$ is the expected value of cell X_{ij} under he assumption that the two streams which are used to construct the contingency table, Φ and Ψ, are independent, and consequently uninformative about each other.

It should not be thought that this necessarily involves first constructing a contingency table, and then operating over this contingency table — we shall see in chapter 8 that for some operators Ω, the collection of the contingency table, and the calculation of the operator over it can be folded into one operation using local, incremental, 'learning rules'.

Structural Operations

Sometimes it is interesting to be able to collect a contingency table which includes information on the dependencies between one (focal) stream, and a number of other (peripheral) streams without having to collect information on all the dependencies which exist between the streams.

For instance, suppose we have a stream Ψ of focal items, with alphabet $\mathcal{A}_\Psi$, and some streams, $\Phi_1, \Phi_2, \ldots, \Phi_N$, of peripheral items with alphabets $\mathcal{A}_{\Phi_1}, \mathcal{A}_{\Phi_1}, \ldots \mathcal{A}_{\Phi_N}$. It is possible to collect the $|\mathcal{A}_\Psi| \times |\mathcal{A}_{\Phi_n}|$ contingency tables $[\Psi, \Phi_n]$ for $n = 1, 2, \ldots, N$. It will be interesting to consider the $|\mathcal{A}_\Psi| \times \sum_{i=1}^{N} |\mathcal{A}_{\Phi_i}|$ contingency table which consist of these N small tables arranged side by side. The advantage of this over collecting the contingency table $[\langle \Psi, \Phi_1, \Phi_2, \ldots \Phi_N \rangle]$ is that there are many fewer cells in the pairwise method outlined above ($|\mathcal{A}_\Psi| \times \sum_{i=1}^{N} |\mathcal{A}_{\Phi_i}|$) than in the case of finding a contingency table of the product stream ($|\mathcal{A}_\Psi| \times \prod_{i=1}^{N} |\mathcal{A}_{\Phi_i}|$). Consequently, much less computer memory is needed to store the former table, and most importantly, a much smaller corpus is needed to find statistically reliable estimates for the values of the cells.

We write $[\Psi, \Phi_1 + \Phi_2 + \ldots + \Phi_N]$ to represent the two dimensional contingency table formed by adjoining the tables $[\Psi, \Phi_1], [\Psi, \Phi_2], \ldots, [\Psi, \Phi_N]$ in this manner. For now, the notation will be kept as a special case, but it is possible to define the '+' operator as an operator between *streams* so that notation may be made more consistent.

5.2 Analysis of the Contingency Table

Once a contingency table, typically a 2-dimensional (or *bigram*) contingency table $[\Psi, F(\Psi)]$ has been collected for some large corpus Ψ, it is necessary then to analyse it to uncover the structure inherent in the statistical redundancy between the co-indexed values of the streams Ψ and $F(\Psi)$. For now we shall be interested how the redundancy inherent in the relationship between Ψ and $F(\Psi)$ can be used to classify the domain in a manner which is likely to be useful to a machine which is attempting to model, or 'learn about', regularities in the domain. I now sketch a procedure for achieving a

hierarchical classification of the alphabet of Ψ.

5.2.1 Classifying the Alphabet of a Stream using Contingency Tables

In order to classify the elements of a domain hierarchically, the following procedure will be used:

1. Let the corpus which is the basis for the classification be Ψ. Choose a set of stream functions to which the structure of the domain which we seek to uncover is sensitive. Let us call these relations $F_i(\Psi)$, $i = 1, 2, ..., I$

2. Collect the contingency tables $[\Psi_{|W}, F_i(\Psi)]$, where $\Psi_{|W}$, as defined in 5.1.2, is the stream Ψ restricted to the set of items of interest, W.

3. The vectors $r_{\psi i} = [\Psi_{|\{\psi\}}, F_i(\Psi)]$ are now taken to be the first approximation of a representation of the word ψ. For each word ψ, there will therefore be I vectors. Let us assume for now that $I = 1$, so there is only one representation per word. These representations correspond to the rows of the contingency table, $[\Psi_{|W}, F(\Psi)]$. It may then be necessary to change the representation of the word ψ to be some function of the entire contingency table. This is the case in some of the 'normalisation' procedures discussed below to factor out word frequency effects from the observations in the contingency table.

4. In order to perform a classification of words, it is necessary to know how similar a given pair of words are. It is to be hoped that we can find some measures of similarity between representations of words which corresponds to the linguistic similarity between them which we want to uncover. Whether this is possible depends on the choice of F. However, it also depends on being able to find a function which can provide a measure of replaceability which is resistant to random variations in representation due to the small sample size.

5. A classification procedure is required to turn the calculated distances between representations into a classification of the lexicon.

5.2.2 Information and Redundancy

Clearly, it is not possible to choose to represent different aspects of an item and apply the techniques discussed above blindly, and still come out with a reasonable structure for the domain we are interested in. For example, if the two intensional operators are statistically independent of each other, then one would expect no significant structure to be derived. For instance, if we were to cross classify the sequence of words in the Bible with the sequence of words appearing in this thesis, then since they have nothing to do with each other, we would expect to uncover no structure whatsoever[2]. Thus it is a necessary, although not sufficient condition that the two representational streams be *informative* of each other, in the sense of Shannon (1948). This notion of 'informativeness' is now more fully considered.

An arbitrary pair of random variables, X, and Y, may be more or less *informative* about each other. The amount to which they are informative of each other depends on how much knowing only the value of one allows us to make better prediction about the other. Independent events are uninformative of each other. For instance, knowing that yesterday I threw a six on a fair die (X) tells me nothing about what the outcome will be today if I throw the die again (Y). On the other hand, if X was the event of drawing a card from a fresh deck of playing cards, and Y were the event of drawing a card from a deck from which one card has already been drawn, then X is informative about Y, since the values of Y and X can't coincide. However, in this case it is clear that the value of X does not give very much information about Y: it reduces the distribution from 52 equally likely possibilities to 51 equally likely possibilities. If, however, X was the weight (in grammes) of a person measured today, and Y was their weight measured tomorrow, then it is clear that although X doesn't predict Y with certainty, it certainly facilitates much better guesses for Y. In corpora, proximal words, such as 'previous word' or 'next word' are highly informative of the current word, and this redundancy is

[2]In actual fact, there will be regularities. For instance, the New Testament mentions Jesus, but the Old Testament does not. The latter half of this thesis mentions "streams", while the first half does not. So there might be a correlation between the word "stream" in this thesis, and the word "Jesus" in the Bible (or other terms which occur only in the New Testament). However, this regularity exists because of a much stronger correlation which exists between words and chapters in this thesis, and between words and books in the Bible.

due to the effects of word order, which are primarily influenced due to the constraints imposed on utterance forms by syntax. This is the sort of redundancy which will be exploited to derive a classification of the lexicon.

Claude Shannon (1948) formalised this notion of 'quantitative information', and linked it to quantitative models of uncertainty, defining 'information' as the expected reduction in uncertainty of the value of a random variable given knowledge of some other random variable's value. Uncertainty was (very plausibly) axiomatised, and a quantity called *entropy*, which is calculated over the probabilistic distributions of random variables, was shown to follow as a uniquely correct measure given these assumptions.

In order to build a classification of the lexicon, we shall compare empirically collected data from the relations **next word**, **next word but one**, **last word**, **last word but one**, and compare this real data to what would be expected if these relations were uninformative given that the frequencies of words is known. This corresponds to assuming the joint distribution has *maximum entropy* (minimal predictive power; maximum uncertainty) given the observed frequencies of individual words. Recall, that a necessary condition for being able to use this relationship to uncover non-trivial structure is that the actual relationship between the streams differs significantly from the one which would hold assuming the streams were uninformative about each other. The contingency table which would result if the two streams were truly uninformative of each other will be called the *null* or MAXENT model. The value of the cell x_{ij} of the MAXENT contingency table is simply $\frac{x_{i+}x_{+j}}{x_{++}}$.

One model of linguistic generation consistent with the MAXENT contingency table (ie. one stochastic model of language generation which would produce the MAXENT contingency table) is to assume that the language stream was arrived at by choosing words entirely at random, but in accordance with their relative frequencies in real text. This is, of course, a very poor model of natural language since all the regularities due to word order are ignored. However, this is a good base model for uncovering facts about syntax since, as noted in 3.2.1, word frequency should be uninformative about the structure of natural language. Consequently the differences between a better model and this model can be used to classify natural language. It is interesting that the insistence that the

relationship between the current word and the previous word be *informative* is a purely statistical necessity, and that this statistical necessity automatically gives rise to a prediction that word frequency will be irrelevant to the structure which will be uncovered. However, previously, linguistic reasons were given for precisely this position.

A better stochastic model of natural language generation is one which generates a contingency table between a focal stream Ψ and a peripheral stream $F(\Psi)$ not in accord with the MAXENT model, but in accord with the actual observation of $[\Psi, F(\Psi)]$. Classifying natural language in this paradigm is a matter of building an ontology in which the observed distributions of the peripheral contexts of the focal items close in the ontology differ from the uninformative null model in similar ways. Once the definition of 'peripheral context' has been decided, all that is left to do is to determine how similarly two focal items differ from the null model of them.

The way this is achieved is by 'factoring out' the null model from the observed contingency table in a process of 'normalisation'. The null model, if true, would give rise to a contingency table with certain, calculable, expected cell values. A statistical test can then be performed between pairs of focal items which, if the null model were true, would give rise to a constant expected distance between all pairs of focal items, but if the null hypothesis were false, gives pairs of focal items which differ from the null hypothesis in a similar way a small distance, and items which differ in very different ways a large distance. All that remains for us to do is to find a statistical test which gives rise to distances which respect our notion of 'similar way', and that this test should preferably be powerful (resistant to the random perturbations in the data).

5.3 Quantitative Measures of Similarity

One formal system which is often used to model similarity is the mathematical concept of a metric. A metric space is a set of individuals, S, together with a function, $d : S \times S \rightarrow \mathbf{R}$, between pairs of individuals and the non-negative real numbers. Small values of $d(x, y)$ indicate that x and y are similar, while large values indicate that they are not. To qualify as a metric, $d(.,.)$ must satisfy certain properties.

$$d(x, y) = 0 \quad \text{iff } x = y \tag{5.5}$$

$$(\forall x, y) \quad d(x, y) = d(y, x) \tag{5.6}$$

$$(\forall x, y, z) \quad d(x, y) + d(y, z) \geq d(x, z) \tag{5.7}$$

(5.7) is the only non-obvious restriction on d, and it is there because $d(x, z)$ can be interpreted as the length of the shortest route between x and z. Were this restriction violated, then supposing that the length of routes is additive (ie. a journey from x to z via y is the sum of the length of the journey between x and y, and the journey between y and z), then the metric d could not be given this interpretation, since then the shortest route between x and z would be longer than a route between x and z via y, which is a contradiction.

Whether this restriction should hold if d is interpreted as a measure of *similarity* between items, rather than the length of paths between them, is an interesting question. However, it is clear that this restriction captures the notion of 'spatial similarity' to some extent, and there would seem to be no pressing reason to change it to something else. However, it will prove useful to consider what properties such a function should possess if it is to interpreted as a measure of similarity rather than as a measure of distance, especially as we have no intuitions about what it means to add similarities together, as we do about adding distances.

Firstly, and in line with the analysis of similarity given in 2.5, any measure of similarity should respect non-deterministic similarity. In terms of this chapter, this means that a stream operation which retains (to a large extent) any statistical regularities which originally pertained within the stream, and which introduces no new ones, should not affect the similarities which pertain between the original items. For example, if every occurrence of a symbol, λ in a stream Ψ is replaced by a new symbol α with probability p, and β with probability $1 - p$, and if this replacement is done at random, then one would expect α and β to be judged as similar. One would also expect that if α is dissimilar to some other symbol, μ, then so too is β. To formalise this argument, I will define the *random replacement operator.*

Definition 5.3.1 *Random replacement operator,* $\mathbf{X}_{\lambda\alpha\beta p}$ *Let* Ψ *be a stream with alphabet* $\mathcal{A}$. *Suppose that* $\alpha \notin \mathcal{A}$ *and* $\beta \notin \mathcal{A}$. *Choose* $\lambda \in \mathcal{A}$. *We define the* random replacement operator, $\mathbf{X}_{\lambda\alpha\beta p}$ *by*

$$\mathbf{X}_{\lambda\alpha\beta p}(\Psi)(i) = \begin{cases} \Psi(i) & \text{if } \Psi(i) \neq \lambda \\ \begin{cases} \alpha & \text{with probability } p \\ \beta & \text{with probability } 1-p \end{cases} & \text{if } \Psi(i) = \lambda \end{cases}$$

Now, suppose d is a similarity measure defined between symbols in the alphabet of Ψ, and $d_{\lambda\alpha\beta p}$ is a similarity measure defined between symbols of the alphabet of $\mathbf{X}_{\lambda\alpha\beta p}(\Psi)$. From the discussion above, we require that

(5.8) $d_{\lambda\alpha\beta p}(\alpha, \beta) \approx 0$

Now, (5.8) effectively captures the content of the statistical replacement test of 3.2.1. The replacement operator, $\mathbf{X}_{\lambda\alpha\beta p}$, defines a stream where the distributions of the contexts of α and β are, for large enough corpora, truly indistinguishable by any means whatsoever (this is the content of the word 'random' in the definition of $\mathbf{X}$, in 5.3.1). However, for real natural language data, it is highly unlikely that any two words have distributions which do not differ significantly. As in the discussion after the definition of the statistical replacement test, while 'red' and 'pink' might be assigned identical syntactic categories, one rarely writes about something being 'shocking red', describes someone as 'pink blooded', or complains about a pink article of clothing fading to red in the wash. Consequently factors related to the meaning of words, pragmatic factors related to how language is used to describe objects in our particular world, and idiomatic usages of words imply that no two words will have identical distributions in large, real corpora of natural language.

Thus, if similarity as judged by some metric, d, is to judge syntactically similar words as similar, then although equation 5.8 is a necessary condition, it is not sufficient, since it will not apply to any two pairs of words in real corpora. One would hope that the differences in distribution between words which are syntactically similar is judged as slight, while those between words with different categories is judged as large.

We might suppose there exists a *syntactic* measure of similarity between words, d_L, which judges as similar syntactically similar items, and as dissimilar pairs of syntactically dissimilar words. Consequently, a desirable property of any *empirical* metric is that it agrees with the syntactic measure. That is,

(5.9) $d(\lambda, \mu) \approx 0$ iff $d_L(\lambda, \mu) \approx 0$

How well various metrics satisfy this property is largely an empirical (post-hoc) matter, and can be given a theoretical analysis only by making unjustifiable assumptions about the nature of the regularities in the corpus (ie. how d_L interacts with bigram statistics). However, for reasons discussed briefly below, measures of similarity based on non-parametric (rank-based) statistics will be preferred. We now discuss a number of candidate metrics.

5.3.1 Similarity Metrics

In statistical theory, there are several such metrics. We shall concentrate on four of them: Spearman Rank Correlation Coefficient; Manhattan (or Canberra) Metric ($\mathbf{L}_1$); Euclidean Metric ($\mathbf{L}_2$, which is closely related to linear correlation coefficient); and Divergence. Their definitions will be given here, and some of their properties discussed.

Spearman Rank Correlation Coefficient: This is defined as follows. Let X and Y be random variables, taking real (or, in general, orderable) values. Let $\{\langle x_i, y_i \rangle | i = 1, 2, \ldots n\}$ be a sample taken from their joint distribution. Let R_i^X be the *rank* of x_i within the set $\{x_j | j = 1, 2 \ldots n\}$ (ie. R_i^X will be 1 if x_i is the largest observed value of X, and so on). Similarly define R_i^Y. The Spearman Rank Correlation Coefficient is based on the following sum:

(5.10) $$\sum_{i=1}^{n} \left(R_i^X - R_i^Y \right)^2$$

If there are ties within the $\{x_i\}$, then a modification of the definition of rank is required. This is achieved by defining the rank of tied elements to be the mean

of the ranks they occupy. Thus if $\{x_i\}$ is $\{2, 2, 2, 1, 1\}$, then $R_1^X = R_2^X = R_3^X = 2$, while $R_4^X = R_5^X = 4\frac{1}{2}$.

The actual definition of Spearman Rank Correlation Coefficient is a linear scaling of (5.10) into $[0, 2]$, followed by translation into $[-1, 1]$ followed by a reflection about 0. This ensures that perfectly correlated data gets a coefficient of 1, while anti-correlated data gets a coefficient of -1. However, (5.10) is the metric on which the correlation coefficient is based, and it satisfies (5.5) – (5.7), so this is what is used.

The Spearman Rank Correlation Coefficient is a robust metric. Hettmansperger (1984) discusses how inference based on rank is often more powerful (ie. less often leads to a type II error, where the null hypothesis is falsely retained) than other forms of statistical inference. Moreover, inference based on rank is asymptotically just as powerful as most powerful tests under the assumption that X and Y have a joint normal distribution. Statistics with high power are desirable in the analysis of natural language data for the purpose of classification, since they are more indicative of an underlying similarity in distribution than low power statistics which may often falsely indicate that words have significantly different distributions when in fact they do not. Consequently, statistics with a high power are more likely to fit the desirable properties (5.8) and (5.9).

Measurements from language cannot be assumed to be normally distributed. The distribution of bigram frequencies is dependent on the distribution of words in the natural language stream, and no assumptions about normality can be made about this. Consequently statistical measures based on rank may have more power than statistics based on the assumption of underlying normality of the variates. It will turn out to be the case for natural language that this measure provides the classification most in accord with linguistic intuitions about the underlying classification.

Manhatten (or Canberra) Metric (L_1): This metric is defined between two vectors, $\mathbf{x} = \langle x_i \rangle$ and $\mathbf{y} = \langle y_i \rangle$. It is defined by

$$(5.11)\quad d(\mathbf{x}, \mathbf{y}) = \sum_i |x_i - y_i|$$

This metric clearly satisfies (5.5) – (5.7). To make it satisfy (5.8), we shall have to divide each of the $\{x_i\}$ by $\sum_i |x_i|$. This 'normalises' the vector $\mathbf{x}$, so that absolute frequency of $\mathbf{x}$ no longer matters. The question of how much this metric satisfies (5.9) is largely an empirical matter. Results suggest that this metric is powerful for artificial data sources, but not so powerful for real data sources.

Euclidean Metric, (L_2): Again this metric is defined between two vectors, $\mathbf{x} = \langle x_i \rangle$ and $\mathbf{y} = \langle y_i \rangle$. It is defined by

$$(5.12)\quad d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2 = \sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} y_i^2 - 2\sum_{i=1}^{n} x_i y_i}$$

To make the metric satisfy (5.8), it is again necessary to normalise the vectors in some way. To see how to do this, we note that this metric has a close connection with linear correlation coefficient which is defined to be

$$(5.13)\quad \rho_{\mathbf{xy}} = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n} (y_i - \overline{y})^2}}$$

where

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

In particular, if the vectors $\mathbf{x}$ and $\mathbf{y}$ are first translated so that the mean value of their elements is 0 (ie. $x_i \mapsto x_i - \overline{x}$), and then *normalised*, so that the sum of the squares of their elements is 1, then, since the mean cannot change from 0 by this normalisation, (5.13) becomes $\sum_{i=1}^{n} x_i y_i$, so

$$(5.14)\quad d(\mathbf{x}, \mathbf{y}) = 2(1 - \rho_{\mathbf{xy}})$$

Now, we know from statistical theory that (5.13) is the optimal measure of association if the data are distributed according to a joint normal distribution. Consequently, (5.14) is the measure which will be used whenever the data approximately satisfies this condition. If the data does not satisfy this condition, then the correlation coefficient based metric becomes unreliable, and as has been mentioned earlier, rank based measures of association become more appropriate to the problem.

Divergence This metric is motivated from information theory, and is related to the minimal expected code length for relaying the contextual value of an individual occurrence of the word **x**. In particular, it is concerned with the difference between predicting the context given that we know the word in question is **x** or that we know the word in question is **y** and the case of predicting context given that all we know is that the word is $\mathbf{x} \vee \mathbf{y}$ (in the first case we can distinguish between **x** and **y**, and in the second we can't).

The *entropy* of a finite probability distribution of a random variable P, $\langle p_i \rangle$ is defined to be

$$(5.15) \quad -\sum_{i=1}^{n} p_i \log_2 p_i$$

This is, in fact, a lower bound for the minimal expected binary description length of the value of P given that we have no additional information about it.

Divergence is a measure between two probability distributions, $\langle p_i \rangle$, and $\langle q_i \rangle$, which measures how far apart they are from their average distribution. We may treat the rows of the contingency table $[\Psi, F(\Psi)]$ as an estimate of the conditional probability distribution of $F(\Psi)(i)$ given $\Psi(i)$, in which the probability $P(F(\Psi)(i) = \phi | \Psi(i) = \psi)$ is proportional to $[\Psi, F(\Psi)]_{\psi\phi}$. In fact, if we write $X = [\Psi, F(\Psi)]$, then

$$(5.16) \quad P(F(\Psi)(i) = \phi | \Psi(i) = \psi) = \frac{X_{\psi\phi}}{X_{\psi+}}$$

If ψ is a focal word (in the alphabet of Ψ), then the representation of ψ is the conditional probability distribution vector, $\langle P(F(\Psi)(i) = \phi | \Psi(i) = \psi) \rangle$, the vector

ranging over the values ϕ of the alphabet of $F(\Psi)$. We denote here (and only here) $P(F(\Psi)(i) = \phi|\Psi(i) = \psi)$ by ψ_ϕ. Therefore, the representation of ψ is $\langle\psi_\phi\rangle$.

Assuming that all pairs of words, ψ^1, ψ^2 are equally likely, which is what must be done to satisfy the statistical replacement criterion which states that frequency of words should not be taken into account when categorising them, we see that $P(\psi^1) = P(\psi^2)$, so the entropy of the context given that the current word is either ψ^1 or ψ^2 is[3]

$$(5.17)\quad H(\phi|\psi^1 \vee \psi^2) = -\sum_{\phi \in \mathcal{A}_{F(\Psi)}} \frac{\psi^1_\phi + \psi^2_\phi}{2} \log \frac{\psi^1_\phi + \psi^2_\phi}{2}$$

where $\mathcal{A}_{F(\Psi)}$ is the alphabet of $F(\Psi)$.

The entropy of the context given that the identity of the word is known is

$$(5.18)\quad H(\phi|\psi^1) = -\sum_{\phi \in \mathcal{A}_{F(\Psi)}} \psi^1_\phi \log \psi^1_\phi \qquad H(\phi|\psi^2) = -\sum_{\phi \in \mathcal{A}_{F(\Psi)}} \psi^2_\phi \log \psi^2_\phi$$

Consequently the expected information which is conveyed by knowing which of ψ^1 and ψ^2 the word is, given that we know it is either ψ^1 or ψ^2 is

$$(5.19)\quad H(\phi|\psi^1 \vee \psi^2) - \frac{1}{2}H(\phi|\psi^1) - \frac{1}{2}H(\phi|\psi^2)$$

and this is equal to

$$(5.20)\quad -\sum_{\phi \in \mathcal{A}_{F(\Psi)}} \frac{\psi^1_\phi + \psi^2_\phi}{2} \log \frac{\psi^1_\phi + \psi^2_\phi}{2} - \frac{1}{2}\psi^1_\phi \log \psi^1_\phi - \frac{1}{2}\psi^2_\phi \log \psi^2_\phi$$

[3]This follows because $P(\phi|\psi^1 \vee \psi^2)P(\psi^1 \vee \psi^2) = P(\psi^1 \vee \psi^2|\phi)P(\phi)$ by Bayes' theorem. From this we see that since $P(\psi^1 \vee \psi^2|\phi) = P(\psi^1|\phi) + P(\psi^2|\phi)$, and $P(\psi^1 \vee \psi^2) = P(\psi^1) + P(\psi^2)$, since ψ^1 and ψ^2 are mutually exclusive values of the stream $\Psi(i)$ as i varies, that

$$P(\phi|\psi^1 \vee \psi^2) = \frac{P(\psi^1|\phi)P(\phi) + P(\psi^2|\phi)P(\phi)}{P(\psi^1) + P(\psi^2)}$$

and applying Bayes' theorem again to the two numerators above, and remembering that $P(\psi^1) = P(\psi^2)$, we see that this becomes

$$\frac{1}{2}\left(P(\phi|\psi^1) + P(\phi|\psi^2)\right)$$

(5.20) is the definition of the *divergence* metric between two distributions. It satisfies (5.5) – (5.7), and (5.8). Again how well it satisfies (5.9) is a matter for empirical investigation. Results suggest that this metric is good for both artificial and real data.

This metric has the interpretation in communication theory of the advantage in minimal expected description length of taking the distinction between ψ^1 and ψ^2 into account if the task is to communicate the context of occurrence of these words over collapsing these two symbols into one symbol.

5.3.2 Normalisation

In order to find a distance metric which meets all the usual properties, it is often useful to preprocess, or 'normalise' the contingency table before applying one of the standard distance metrics as described in the last section. The idea behind preprocessing is to transform the table so that linguistically salient redundancies are more readily uncovered by the standard metrics described above.

Now, all of the above metrics satisfy (5.5) – (5.8), except $\mathbf{L}_1$ and $\mathbf{L}_2$, which do not satisfy (5.8). Normalisation can be used to preprocess the contingency table for two reasons. Firstly, to ensure that the metric defined over rows of the contingency table is in fact a metric over words satisfying the replacement criterion (ie. it satisfies (5.8)), and secondly to help the metric meet the desirable property (5.9). In order to achieve this it is often useful to transform the values of the cells of the contingency table so they are not estimates of the actual bigram statistics, but rather the differences between estimates of the actual bigram statistics, and what they would have been had there been no effect due to word order.

In the first case, then, normalisation is used to force the metric to satisfy (5.8), and in the second case to make the metric more sensitive to syntactic structure as defined by (5.9). The latter is achieved by changing the values of the cells by the Ω_E normalisation operator defined in (5.4).

5.4 Hierarchical Classification

Having defined a measure of similarity between items based on empirically observed statistics of the contexts in which they occur, the question of how to use these measurements to deduce structural information about language presents itself.

We want to assign a topology to the set of focal items which respects the observed similarities which have been statistically determined. There are many candidate topological spaces in which the focal items may be placed, including all the usual spaces — n-dimensional vector spaces; Tori; Circles; and so on. For the purposes of natural language, and many other domains, a particularly interesting topological space to embed the alphabet of the focal stream into is a hierarchical taxonomy, or tree. Such a hierarchical taxonomy is called a *dendrogram.*

One algorithm which constructs dendrograms, which was first defined by Sokal & Sneath (1963), can be defined as follows.

```
START Put each item in a cluster on its own.
WHILE(there is more than one cluster remaining)
BEGIN
    Find the closest two clusters.
    Create a new cluster containing those two clusters as branches
    Delete the two clusters from the list of remaining clusters.
END
```

There is, of course, the problem of finding the distance between clusters, given that we only know the distance between items. Several methods have been tried. Let X be the set of elements of the alphabet in one cluster, and Y be the set (disjoint from X) of elements of the alphabet in the other set. Possible definitions of the distance between the clusters X and Y include:

MIN: The distance between two clusters is taken to be the minimum distance between

the elements of the clusters. Thus,

(5.21) $d(X,Y) = \inf\{d(x,y)|x \in X, y \in Y\}$

MEAN: The distance then becomes the mean distance between the leaves of the two clusters.

(5.22) $d(X,Y) = \frac{1}{|X||Y|} \sum_{x \in X, y \in Y} d(x,y)$

AMALGAM: The distance between two clusters is the distance between two points formed by some combination of the values of the contextual statistics of the leaves of the clusters. For example, if the contingency table is $[\Psi, \Phi]$,

(5.23) $d(X,Y) = d(\frac{1}{|X|} \sum_{x \in X} [\Psi_{|\{x\}}, \Phi], \frac{1}{|Y|} \sum_{y \in Y} [\Psi_{|\{y\}}, \Phi])$

This corresponds to the distance between the two labels X and Y where all the items which are one of the leaves of X are replaced in the corpus by a label X, and all the items in the corpus which correspond to one of the leaves of Y are replaced by the label Y.

The choice made of the distance between clusters makes a difference to the structure of the resulting dendrogram. In particular, the MIN distance measure favours large clusters with new elements being added one at a time, rather than in clusters. This is because it is relatively more likely that an arbitrary element is closest to an element in the large cluster (than, say, to a singleton, all other things being equal), and this gives a dendrogram with a flatter structure than the other choices, with many long lists of elements.

Since the interpretation of the value of a cell contingency table is the number of items in a corpus which fit a particular description, the sum of two cells corresponds to the number of items fitting either the description associated with the first cell, or the description associated with the second cell. Given this, the interpretation of the sum of two rows is the contextual statistic associated with the cluster containing the two items concerned. This clearly extends to the sum of an arbitrary number of rows, and arbitrarily large

clusters. Because of this, possibly the best motivated measure for forming clusters is the AMALGAM measure, but this carries with it additional computational complexity in the form of needing to keep a record of the original statistics, as well as the distances between items. Also, if the contingency table has already been normalised, it would seem necessary to repeat this normalisation every time a new category is formed, thus, depending on what the method of normalisation actually is, possibly necessitating a computationally expensive recalculation of the distances at every stage in the construction of the dendrogram. A further criticism of this metric is that it is, at least according to some metrics, possible for the distance between two large clusters to be smaller than the distance between the daughters of the clusters. Thus it is possible using this definition to get non-monotonic clusterings. Consider, for instance, Euclidean distance on the plane, and three points on the vertices of a unit equilateral triangle. The distance between all pairs of points is 1 unit. However, after two of them have been clustered according to the AMALGAM metric, the distance between the third point and the cluster is now $\frac{\sqrt{3}}{2} \approx 0.87 < 1$. According to both the mean distance, and minimum distance metrics, however, the distance is 1 unit.

The choice used in the dendrograms presented here will, unless otherwise specified, be the MEAN measure. This measure is commonly used, produces fairly richly structured trees, and on small problems was empirically found to be at least as good as the AMALGAM measure at extracting known structure from data.

5.4.1 Problems with Numerical Taxonomy

I must mention that although it is a technique widely applied within the social and biological sciences, one problem with this approach to producing a hierarchical classification of a domain is the lack of a coherent statistical interpretation of the resulting dendrogram. Both the choice of metric measuring the dissimilarity between lexical items, and the choice of measure for calculating the dissimilarity between clusters of lexical items are, to some extent, arbitrary. For example, if more lexical items are added to the list to be classified, it is not in general true that the classification of the original items will be consistent with the new classification. The only exception to this is if the measure

used to calculate the distance between clusters while constructing the dendrogram is the MIN measure above. The resulting classification, therefore, should be thought of as a means of representing the distance between items in a tree, and as such is just a means of presentation of data.

This thesis, however, uses this technique as a component of the exploratory analysis of a stream of data with (relatively) unknown structure. In this context, a hierarchical classification, no matter how it is derived, might be very informative about the underlying structure of the domain being investigated. Whether this is true or not is an empirical matter. One goal of the exploratory component in a machine learning system is to derive an *approximate* classification of the domain which can later be refined, or used to provide a more adequate model of the domain (in this case the syntax of natural language) than, for instance, the null model described above in the discussion on information and entropy.

One means of making the mapping between measured distances and dendrograms more statistically well-motivated is by exploiting the fact that there are classes of naturally definable metrics we can derive from a given tree. Essentially, it can easily be shown that any mapping, D, between the nodes of a tree and the real numbers which is strictly monotonic with respect to the 'ancestor' relation, and maps each leaf to 0, defines a metric, d, where $d(x, y) = D(x \vee y)$, where $x \vee y$ is the most recent common ancestor of x and y. The problem is then reduced to finding a means of finding that tree which gives rise to a metric most similar to the original measurements of similarity. This problem has been extensively studied in the literature on *multi-dimensional scaling* (Bechtel 1976), and notions from that paradigm, such as *stress*, carry over naturally to a multi-dimensional scaling approach to defining a dendrogram as 'that tree which minimises stress'. I will not attempt to elucidate this point further here, however.

5.4.2 Representing Dendrograms

In all the figures which follow, the dendrograms shall be drawn as a horizontal binary branching tree, with the items of interest on the right hand edge of the dendrogram, and the root of the entire cluster displayed on the left hand edge. The distance of each

mother node from the right hand edge is proportional to the distance between its two daughter clusters as calculated by the dendrogram generation algorithm given above. An explicit scale will not usually be given, since it is usually only the structure of the tree which is interesting, and not the precise values of the correlations between the items. Unless otherwise specified in the text underneath the figure, all of the dendrograms are presented without alteration from the output of an automatic dendrogram generation program.

5.5 Discussion

A framework has been defined which facilitates the unsupervised investigation and derivation of structure in many domains from statistical redundancies within an indexed corpus. To apply it, one needs an indexed corpus, some stream functions of this corpus, a similarity metric, and possibly a normalisation procedure. Once these have been defined, the procedure to produce a classification of the items within the domain is automatic.

However, the question of what stream functions to use so that interesting structure can be derived, poses non-trivial problems. We shall see later that different sorts of structure can be uncovered by choosing different stream functions. There are two separate approaches to finding stream functions which facilitate the recovery of interesting structure, and these correspond to the innate (knowledge driven)/ unsupervised (data driven) dichotomy. If an innate view is taken, then it might be assumed that although the learning agent doesn't necessarily know what structure it will find, it knows where to look for this structure. This corresponds, in this model, to having a pre-specified set of peripheral stream functions. The alternative view is that, to some extent, that regularities found early on in the data determine what stream functions will be used to uncover structure.

The approach presented here is an amalgamation of the two approaches described above. First, a preliminary survey is made to find the set of the 150 most common words, W, and then a set of stream functions is defined which make reference to the observed com-

mon words, but which are otherwise "innate" (that is, specified prior to the application of the algorithm). This set of functions is typically $\Delta_i(\Psi)_{|W}, i = \pm 1, \pm 2$, where Ψ is the original corpus. It is entirely conceivable that procedures can be found which allow the definition of the stream functions to be changed according to the regularities found within the data. If it were possible to give an empirical definition of 'degree of clustering', for instance, it might well be possible to find a procedure which found stream functions which provided an optimal clustering of the items.

A more detailed description of how this technique is applied in a variety of artificial and real cases will be given in the next chapter.

Chapter 6

Lexical Experiments

This chapter, and the next one, detail the experiments in lexical classification performed on some corpora from various domains. This chapter describes the computational tools which are used to apply the theory of chapter 5 to some artificial and real streams, or corpora. When dealing with real corpora, it is necessary to pre-process the data to convert it into a stream of data most likely to yield to the techniques in classification which are applied to it. Such manipulation includes the removal of parts of the corpus related not to natural language, but rather to computational housekeeping of files, and increasing the apparent size of the corpus by removing irrelevant distinctions, for instance by identifying upper and lower case characters[1].

In particular, this chapter deals with three artificial and three real domains. First, we consider the redundancy between two streams which bear a stochastic relationship to each other which respects a certain topological structure, as was the case with weights and extensions of the spring in the example presented in section 2.7.1 (a linear order), and see how this relationship can illuminate the underlying topology of the domain. Secondly, we consider some simple artificial data due to Elman (1990), and consider what the resulting classification shows in this case. Thirdly, a simple stochastic context free grammar is defined, and output from this analysed. In this case, the hierarchical analysis

[1]This increases the size of the corpus because if the upper/lower case distinction is irrelevant to the category of the word, then a fixed sized corpus will have more tokens of a particular type when case folding has been done than before.

is more illuminating than that of Elman's simpler data. Fourthly, we classify letters taken from a stream of words in Newsgroup articles, and show how a distinction between vowels and consonants is easy to derive. Fifthly, we do the same for phonemic data hand transcribed from the Lund corpus, and finally we derive a hierarchical classification of words from Newsgroup articles, and consider how close this is to orthodox classifications of the lexicon.

6.1 Materials

The materials used for these experiments fall into two classes: (a) The corpora used for the data, and (b) the computational hardware and software needed. The following two subsections discuss these in turn.

6.1.1 Corpora

Several corpora were used for these experiments. Most of the data for the natural language section of the experiments came from USENET Newsgroup articles. More precise details are given within the descriptions of the particular experiments, but a general description of newsgroup data will be given here. USENET is the NSF Internet's world-wide computer bulletin board, connected to many institutions from academia and the business community. Anyone connected to it can, in principle, post an article to any one of over 200 newsgroups, each of which is a forum for discussion in a certain field (for instance, "rec.games" is a newsgroup for the discussion of games, while "sci.math" is one for the discussion of issues relating to mathematics). News articles are posted to USENET by a program which 'stamps' each article with an identification, and various administrative labels. There are several points to note about USENET newsgroup articles as a corpus.

- Often words are misspelt, and often non native speakers of English write articles, leading to undoubted grammatical errors in what is written. Thus the articles are an inherently syntactically 'noisy' source for a corpus.

- Most of the articles originate in the US, which means that American English predominates.
- The largest volume of articles concern issues relating to computers. This biases the vocabulary observed overall within the newsgroup articles to include far more computer based words than in English texts as a whole.
- Often, people 'reply' to others' postings, and this leads to some of the text of the original message being included in the reply. This leads to articles containing verbatim repeats of parts of other articles. In some circumstances this might bias the observed contextual statistics.
- People often include their name, address, and perhaps a quote, at the bottom of the articles they post in order to identify themselves, and to personalise the message. This is usually the same for all the articles they post, and this 'signature' is a significant amount of the text of most articles. Consequently contextual statistics might be biased by statistically untypical signatures.
- Some articles are written in foreign languages. This is currently a small (less than 2%) proportion of the data.
- Some articles are encrypted using a simple replacement cipher called "Rot-13". This applies to a very small proportion of articles.

6.1.2 Computational Tools

In line with the UNIX philosophy, a modular approach was taken to the computational implementation of the ideas presented above. A number of small 'modules' or **tools** were built, and connected together using the operating system, to provide a powerful and flexible computational framework for the experiments detailed below. The tools fall into four main classes:

1. **Tools to generate and manipulate the corpus.** This involves tools which scan the directories where news articles are stored, locating unanalysed articles, strip them of their administrative headers, split the input into words, or letters,

convert the characters to lowercase, replace certain items with other items (for instance words with their classes), and generate new items.

UNIX's **find** command was used to scan all the directories which contained news articles to provide their filenames. These articles were then scanned by a text processing program which maintained a reference of whether the article had already been read (by scanning a cross-reference field in the header), in which case the article was discarded. Otherwise a number of heuristic tests were used to 'clean up' the article before further processing. First the header was stripped off the article. The header was defined to be all the text before the first blank line of the text body. Secondly, all lines beginning with the characters ">", "|", and "}" were discarded. This is because these symbols indicate that the text on the line is quoted from another article which has, presumably, already been processed. Were this not done, the apparent size of the corpus (in terms of words processed) would overestimate the true size of the corpus, and therefore quoted corpus sizes would be misleading. Thirdly, if a line began with the character string "--", "**", "##", or "==", the rest of the article was discarded. These characters are a good indicator that the rest of the article is a "footer" giving personal details of the sender's name and address, which should not be included both for similar reasons to why quoted text should not be included, and because the text is very untypical English, so might bias the statistics collected. The rest of the text was broken into "clauses", defined to be that which is between punctuation and contains only alphabetic characters. If a non alphabetic character was detected in a phrase, the phrase was deemed to be noisy, and ignored. All text was also converted to lower case before processing. This processing tool was called **grab**.

There are also tools which take a sequence of items from the corpus processing tools above, and generate new sequences of items consisting of sequences of other items. For instance, the sequence "the man said that he was a linguist" might be converted to "Det Noun Verb That Pron Verb Det Noun", provided that it was given rules such as "the $\rightarrow$ Det", and so on. Also, this tool could "glue" adjacent items together to form a new stream of conjoined items, such as "the-man man-said said-that ...", or "t h e SPACE m a n SPACE s a i ...". This tool was called

manipulate.

Using these tools, the raw data could be flexibly and efficiently extracted and manipulated from the news directories on a daily basis.

2. **Tools to generate the statistics.** The philosophy used in generating a contingency table from a corpus is in line with the ideas presented in section 5.1. That is, a corpus is considered as one long sequence, with an index which represents the "current" item. Statistics are defined relative to the current item (such as "previous item", or "the concatenation of the current item with the next item"), and a contingency table is just the observed co-occurring values of pairs of statistics as the index moves through the corpus. Computationally, this is implemented by having a program which takes as input a sequence of items (either from the corpus processing tools above, or other programs which generate a sequence of items) and outputs a sequence (to be thought of as an unordered set) of ordered pairs (or in general n-tuples) corresponding to the value pairs of the relevant statistics. Thus it might take the stream "the man said ...", and output "(the man), (man said), ...". The program to do this was called **n-gram**, and can be thought to be generating ordered n-tuples of values of statistics.

3. **Tools to generate and manipulate the contingency tables.** This is relatively straightforward. Dimensions of a particular table have to be identified with the values of various statistics, and this tool takes the ordered pairs (or, in general, n-tuples) output by the tool **n-gram** above to increment the appropriate cell of the table. The tool which did this was called **contingency**. For reasons of resource limitations, it is important that this tool works with contingency tables stored in compressed format.

   There are also utility tools which allow the peripheral dimensions of contingency tables to be algebraically 'added' as described in section 5.1, tools which allow conversion between different formats (binary integer and binary floating point representation, and between binary representations and ascii representation), and programs which extract parts of a contingency table, and do simple algebraic manipulations on rows.

4. **Tools to analyse the contingency tables.** The distance between the focal items has to be calculated in many ways as described in section 5.3. These distances then have to be used to generate the dendrograms, again in several possible ways (see section 5.4), and finally there are computational tools to generate graphical output from the dendrograms themselves.

In implementing these tools, a number of standard UNIX tools have been used: **awk, compress, uncompress, find**. All the tools I developed were written in the 'C' programming language. For reasons of efficiency, no standard statistical analysis package was used. A command such as

```
find /usr/spool/news -mtime 1 -print | grab | manipulate -l | \
n-gram current next | contingency -t news.tab -y focal.lst -z context.lst
```

updates the contingency table **news.tab** whose rows correspond, in order, to a list of words in **focal.lst**, and whose columns correspond, in order, to words in **context.lst**. The relation being recorded is ⟨**current-word, next-word**⟩, and this is generated from the corpus generating tools **find, grab, manipulate** by **n-gram**.

6.2 The Experiments.

The method employed to perform most of the experiments is fairly straightforward, and requires little explanation. In each case below, a focal and peripheral dimension were decided on (this varies from case to case and is detailed below), and a large corpus was analysed (the size of the corpus in each case is detailed below). These decisions were converted into files and instructions for the various programs detailed above, and these in turn were executed on networked machines with a Sun-4 SPARC architecture.

Most of the decisions made are about

1. What corpus to use. This is problem of constructing the corpus, as a stream Ψ.
2. What items to consider in the cluster analysis. This decision was usually made by using only the most frequent items in the domain. Typically, then, this will be

$R_W(\Psi)$, where W is the set of N most common symbols in Ψ for some computationally convenient N.

3. What streams to use as peripheral streams. This is an important decision, since the redundancy between these streams and the focal stream should capture as much of the sequential redundancy in the focal stream as possible. Usually, the peripheral streams will be shifted versions of the focal stream, restricted to a small set of symbols. One common one used here is **previous word if it is in the 150 most common symbols of** Ψ, which is defined in the notation of chapter 5 as $(R_C \circ \mathbf{\Delta}_{-1})(\Psi)$, where C is the set of 150 most common items in Ψ.

4. What metric to use to find the similarity between lexical items. Typically, many were tried, and reported.

6.2.1 Foundational: Linear Order.

In order to illustrate the technique, first I show how some aspects of the known structure of a domain can be judged from non-deterministic information about it. Consider two random variables, each taking one of a hundred values. The values are thought to be a quantisation of the real number line between the value of 10 and 20. The observations are a non-deterministic function of the original value (in this case, the original value plus a random variable, again quantised.) This can be written in the notation introduced earlier,

$$[\Psi, \Phi] : \Phi(n) = \frac{1}{20}\left(\Psi(n)^2 + \mathcal{N}(0, 40)\right)$$

Where $\mathcal{N}(\mu, \sigma^2)$ is a normally distributed random variable with mean μ and variance σ^2.

Where in this case, the symbol "+" is to be interpreted numerically. That is, the second variable is obtained from taking the square of the first variable, adding a normally distributed random variable with mean 100 and variance 40 to it, dividing by 20, and quantising the resultant value so that it takes one of the values of the peripheral statistic.

Focal Items: Numeric values between 10.0 and 19.9 in units of 0.1.

Focal Stream, Ψ: Stream of uniformly distributed random numbers in the range 10.0 – 19.9.

Peripheral Items: Numeric values between 0.0 and 30.0 in steps of 0.2.

Peripheral Stream, Φ: $\Phi(i) = \Psi(i) + N$, where N was generated as a normal random number with mean $\mu = 0$, and variance $\sigma^2 = 40$.

Corpus: None. The computer directly generated co-dependent statistics related by the fact that the second (y) was

$$y = \frac{1}{20}\left(x^2 + \eta\right)$$

where η is the value of a random variable distributed as $\mathcal{N}(0, 40)$, independently chosen each time. y was truncated to 1 decimal place.

Best Measure: L_1 after normalisation.

The resulting statistics show that there is a (mostly) monotonic relationship between the numerical distance between focal values, and the degree of correlation between them. The "best" distance measure for this artificial problem was found to be simply *Manhattan metric*, or the sum of the absolute differences between the values of the peripheral variable of items. For instance, the following list shows the the order of the nearest neighbours of a given focal item for both the L_1 and Spearman metrics.

L_1: 15.0: 15.0, 14.9, 15.1, 14.8, 15.2, 14.7, 15.3, 14.6, 15.4, 14.5, 15.5, 14.4, 15.6, 14.3, 15.7, 14.2, 15.8, 14.1, 15.9, 14.0, 13.9, 16.0, 13.8, 16.1, 13.7, 16.2, 13.6, 16.3, 13.5, 16.4 ...

Spearman: 15.0: 15.0, 15.1, 14.8, 14.9, 15.3, 14.5, 15.2, 15.5, 14.4, 14.7, 15.4, 14.6, 15.6, 14.3, 15.8, 15.7, 14.2, 14.0, 13.9, 14.1, 15.9, 16.0, 13.8, 13.7, 16.1, 13.6, 16.2, 16.4, 16.3, 13.5 ...

The resulting dendrograms (figure 6.1) show how we might "coarsen" the quantisation of the domain without losing too much of the domain's "structure". In this case, the way this is achieved is by clustering together focal items which are numerically close, rather than those which are distant. This corresponds to the fact that the mother nodes near the bottom of the tree have atomic descendants which tend to be numerically closer than those of mother nodes higher up the tree. The fact that the L_1 metric-based dendrogram happens to keep the order intact (as we look down the list of leaves in figure 6.1) is an artifact of the fact that it reads the names of the focal values in the correct order, and the algorithm finds no reason to transpose any values. The fact that the leaves of the tree *can* be drawn in the correct numerical order is not an artifact of the dendrogram displaying algorithm, and is a strong reason for preferring L_1 to other metrics.

It should be noted that had the value of the peripheral dimension depended *deterministically* in a 1-1 mapping of the focal dimension onto the peripheral dimension, the techniques used here could not be used to learn about the structure of the domain. This is because no priorly assumed notion of *similarity* is assumed — this notion is deduced by observing the non-deterministic way the peripheral distributions depend on the focal items.

6.2.2 Analysis of Elman's Data

As discussed in chapter 1, Elman has performed unsupervised classification experiments on simply structured language-like 'sentences' generated using very simple finite state automata using recurrent neural networks. From a small lexicon, he attributed each item a small set of categories, and provided simple templates for the combination of these categories into strings. From these templates of strings of categories, he derived a set of strings of words which he used as his corpus. Some examples of these strings are given now:

- rock move
- man smash plate

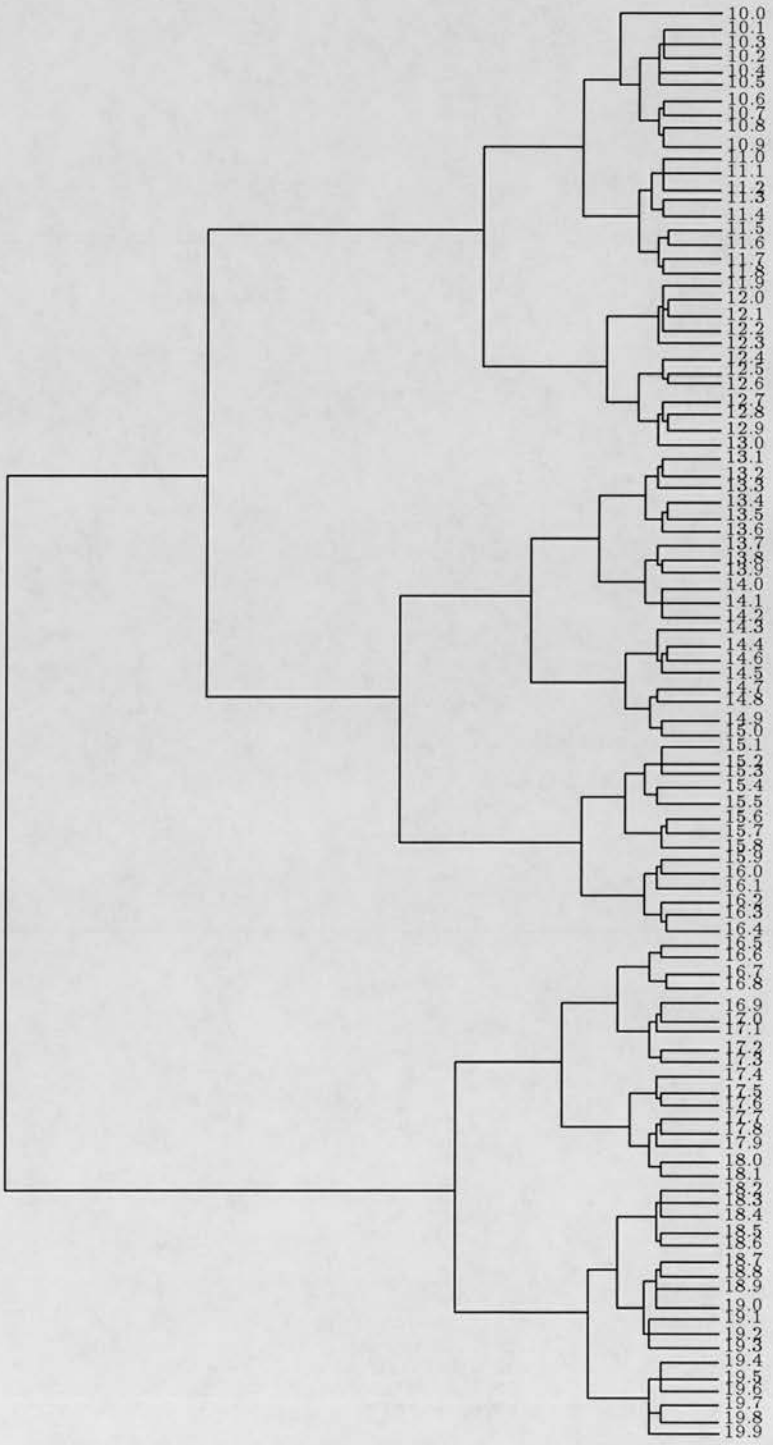
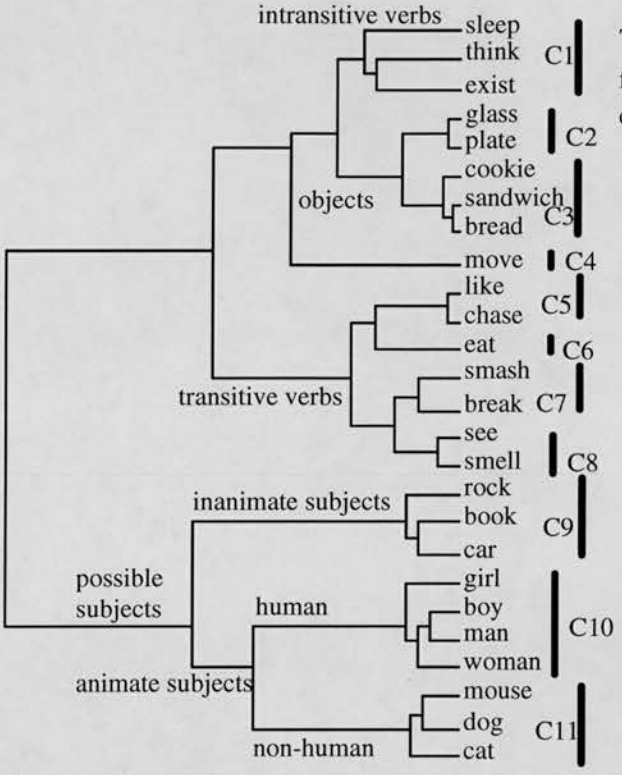


Figure 6.1: This figure shows the dendrogram resulting from clustering the distances calculated between focal items using the streams described in the text.



The dendrogram of the clustering resulting from analysing Elman-style data which contains the following paradigms:

C10 C1 (man sleep)
C10 C7 C2 (woman break plate)
C9 C4 (rock move)
C11 C4 C9 (mouse move book)
C10 C8 C9 (girl smell rock)
C10 C6 C3 (man eat cookie)
C11 C6 C3 (cat eat sandwich)
C11 C5 C11 (mouse chase cat)
C10 C5 C10 (girl like boy)

It is apparrent that the resultant dendrogram is consistent with the extant equivalence classes (thick, solid black lines). The only useful higher-order category is [C10 C11] which compresses C10 C6 C3 and C11 C6 C3 to [C10 C11] C6 C3.

Figure 6.2: This figure shows the dendrogram resulting from analysing $[\Psi, \mathbf{\Delta}_i(\Psi)]$, for $i = \pm 1, \pm 2, \pm 3$. Ψ was defined to be 4000 lexical items generated by concatenating (without any 'stop' markers) random instantiations of the paradigm. The contingency tables were adjoined into a large contingency table, which was then normalised as described in section 5.3.2, and the Spearman rank correlation coefficient was used to calculate the distances between representations. No random noise was added to the contingency table before analysis.

- boy eat cookie
- mouse move rock
- woman sleep

In general, words are marked for whether they are nouns or verbs (transitive or intransitive), whether they are animate or inanimate, whether they are human, whether they are foodstuff, and so on. Further details are given in figure 6.2.

One observation, and possible criticism of figure 6.2 is that there is no clear division between verbs and nouns. This, however, is a criticism of the *data*, and not the analysis process. As commented in the figure, the paradigms from which the data is generated do not really support the "Subject-Verb[-Object]" analysis they might appear to if

they are considered as a subset of an idealisation of the English language. In order to motivate a linguistic distinction between verbs and nouns, it is necessary to show that the distinction is useful in explaining linguistic distributional data in many paradigms. The only higher order category which can usefully be defined is the category [C10 C11], corresponding to *animate subject*, and this is only useful in compressing two of the 9 paradigms into one combined paradigm.

Indeed, by changing the words, but keeping the paradigms, it becomes clear that these paradigms could model an idealisation of a different subset of English which clearly does not support the "Subject-Verb-Object" analysis. Instead of "sleep, think, exist" use attributive adjectives in verb phrases such as "happy, sad, angry", and we still get understandable "sentences". If the paradigms are extended to include other, equally plausible paradigms, the data gets classified slightly differently. Figure 6.3 shows how adding the following paradigms changes the classification:

C10 C5 C11 (boy chase dog)

C11 C5 C10 (dog like girl)

[C2 — C3 — C9] C4 (cookie move)

[C10 — C11] C8 [C10 — C11 — C9 — C3 — C2] (mouse see plate)

Now, when clustering is performed, there is a primary division between nouns and verbs.

Thus one should always be careful when using the method of fragments statistically. Just because a certain formal system *can* be given a linguistic interpretation does not mean that any derived structure can be extrapolated to the fragment of natural language which the fragment seeks to model. Often, there will be many possible linguistic interpretations of a fragment such as the one Elman used, and often these will have different linguistic structures within the larger framework of a complete theory of language, and researchers should avoid concluding anything at all about natural language from behaviour on such highly artificial data. However, with such artificial formal systems, it is possible to define terms such as 'consistent categorisation', which respects the categorisation of the formal system.

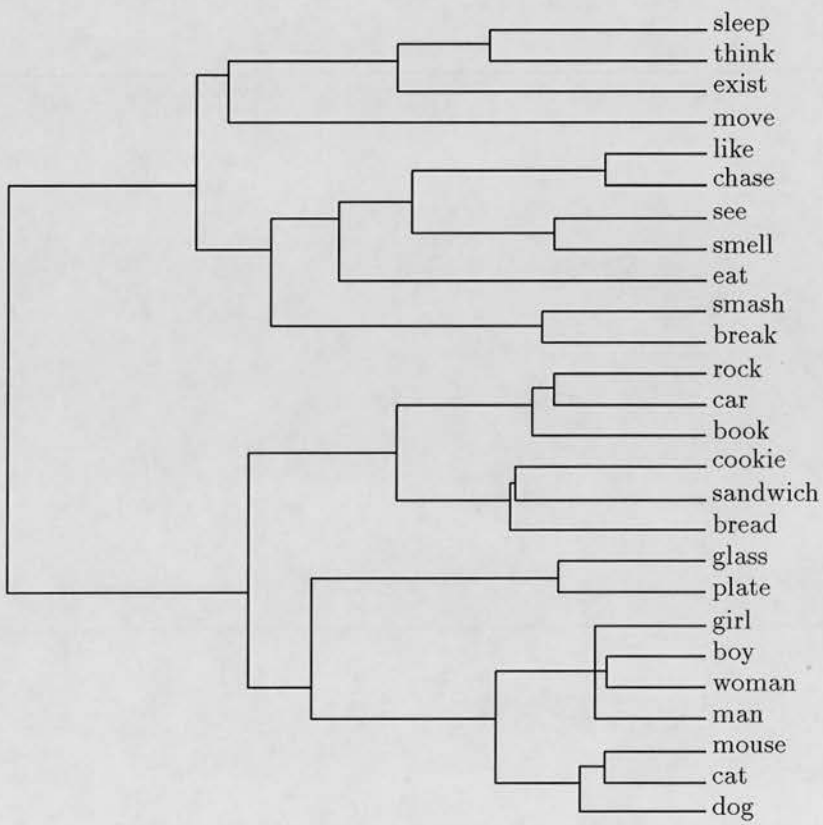


Figure 6.3: The clustering obtained by the same procedure as above, but using the expanded set of rules given above.

6.2.3 Syntactic Categories from an Artificial Grammar

A better artificial approximation to English can be obtained from a stochastic context free grammar (SCFG), which is a context free grammar where the various expansions of non-terminals have a certain probability of being chosen, and this probability is independent of any contextual information. A very simple example is given below.

1. $S \rightarrow aSb \qquad (p = 0.7)$
2. $S \rightarrow ab \qquad (p = 0.3)$

This CFG generates (or recognises) strings of the form $a^n b^n$. This stochastic context free grammar generates strings with the following probabilities:

String	Probability
ab	0.3
$aabb$	0.3×0.7
...	...
$a^n b^n$	$0.3 \times (0.7)^{n-1}$

Since

$$\sum_{n=0}^{\infty} 0.3 \times (0.7)^n = \frac{0.3}{1 - 0.7} = 1$$

the probability of generating a finite string with this grammar is 1. This will not always be the case in context free grammars where a non terminal can introduce more than one non-terminal, and it will be necessary to find probabilities which make the probability of generating a finite string 1. This is, in general, easy, since a sufficient condition is that the expected values of the length of the expansions is finite, and this can be checked by solving a set of linear equations, and checking that the values of the solutions are positive.

Context free grammars can account for many more paradigms of language use than Elman's simple finite-state grammar can. Gazdar et al. (1985) have shown how, using GPSG, a context free grammar can be used to account for a very large number of

linguistic phenomena. The actual SCFG used to generate the corpus is given in the appendix (B), and consists of about 20 non-terminal symbols, 30 non-terminal rules, and about 100 terminal symbols.

This gave rise to a corpus, or stream, of artificially generated text, bearing a passing resemblance to English. A few examples, taken at random, are given below

- her happy hamster deliberately miaows.
- all rabbits would really reject all people that like the clock.
- the book on the happily carefully deliberately big computer could carefully give steve my book.
- my bag rejects her hat.
- computers against her happy rabbit of the happy people will deliberately sing.

The sentences were concatenated in the order they were generated, thus defining a *stream*, Ψ_{SCFG}, the alphabet of this stream being the terminal symbols of the SCFG, which are the words in the above examples. This was analysed using the contingency table statistic

$$[\Psi_{\text{SCFG}}, \Delta_{-1}(\Psi_{\text{SCFG}}) + \Delta_1(\Psi_{\text{SCFG}}) + \Delta_{-2}(\Psi_{\text{SCFG}}) + \Delta_2(\Psi_{\text{SCFG}})]$$

The resultant dendrogram of items, is displayed in figure 6.4. Again, it is interesting to note that in this artificial case, the L_1 metric seems to do better than the Rank correlational metric (although only the L_1 dendrogram is actually shown here for reasons of space).

Turning to the dendrogram, it is perhaps not surprising that this method does so clearly show (most of) the structure of the DCG's lexicon: After all, each plural ditransitive verb should be interchangeable with every other one, and distributed identically, so if there is any significant difference in distribution it must be an artifact of a small sample size, so any significant difference at all in distribution of the classes (which there

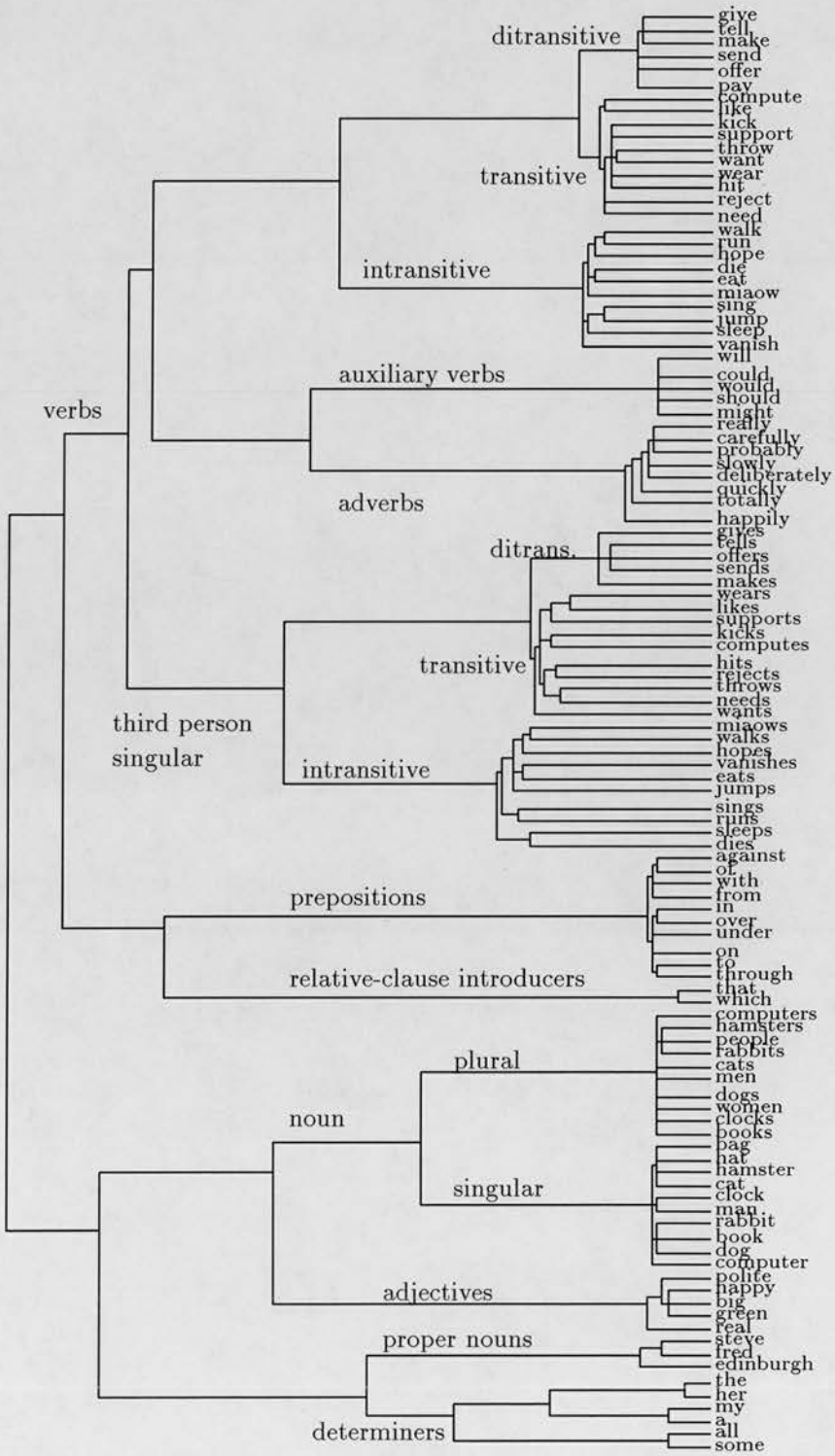


Figure 6.4: This figure shows the dendrogram resulting from an analysis of a corpus of data generated by the SCFG listed in appendix B

should be to meet the expressiveness criterion described previously) will mean that this technique will succeed in correctly classifying the lexicon. However, it is interesting in this context that a difference can be detected between transitive and ditransitive verbs, since the disambiguating context (the second noun-phrase) is typically a considerable distance from the verb. However, short verb phrases (e.g. verb proper-noun proper-noun) are apparently common enough to cause significant differences in the statistics collected. However, this is the last category to become apparent by this distributional analysis — the difference in representation between transitive and ditransitive verbs is so small that they are liable to be confused for small corpus sizes.

What is more interesting, however, is how much more structure than just identical lexical classes is revealed. This is especially clear in the verb subcategory, where not only are the verbs split into their particular maximally specific classes, but also the structure within these classes is evident, the verbs first being split according to whether they are third-person-singular (subject agreement), then according to how many compliments they take (structural agreement). In fact, the resultant dendrogram is very similar to the dendrogram which results from the analysis of real natural language texts, with adjectives clustering close to nouns, and prepositions being close to conjunctions and relativisers.

Corpus, Ψ: concatenation of 200,000 sentences generated from the SCFG, each sentence being followed by the special symbol PERIOD, and preceded by the symbol START.

Focal Items: The terminal symbols of the grammar.

Focal Stream: Ψ restricted to the focal items.

Peripheral Items: The terminal symbols plus START and PERIOD. This is the entire alphabet of Ψ.

Peripheral Stream: $\Delta_i(\Psi)$, for $i = \pm 1, \pm 2, \pm 3$.

Representation: Normalised representations from $[R_F(\Psi), \sum_{i=-3}^{3} \Delta_i(\Psi)]$. No noise was added to the contingency tables.

Best Measure: $\mathbf{L_1}$.

6.2.4 Orthographic Clustering

We turn our attention now to real data. Firstly, we shall consider orthographically represented data from a large corpus of news articles, written in English, taken from the USENET news service. Thus the corpus will be considered as a stream of letters, or characters if space and punctuation marks are included.

Linguistically, letters and punctuation have a relatively complicated mapping to sounds. Sounds, in phonology, can be modelled by phonemes, and these classified according to their sonority, which is an order on phonemes which accounts, through the *sonority principle*, for the breaking of words into syllables. Highly sonorous sounds are typically vowel-like (a as in bat, o as in strong, ee as in beech, y as in syllable etc.), where sounds with a low sonority are consonant-like (ck in back, t in fat, p in stop, etc.). Some sounds have an intermediate sonority, examples include zz, as in 'buzz', ng as in 'sing', y as in 'yacht', and so on. Although the sonority principle, which says that language is broken up into syllables, syllable breaks appearing at local optima in sonority of the surrounding sounds, is quite useful in explaining the distribution of sounds in language[2], the mapping between letters and sounds is irregular. Consequently, regularities due to sound might be obscured by this mapping. Fortunately, the vowel-consonant distinction in English is so strong, vowels so typically being associated with highly sonorous sounds, and consonants with un-sonorous sounds, that this distinction is still predominant.

Corpus: Slightly over $2\frac{1}{2}$ million newsgroup characters, converted to lower-case.

Focal Items: The lower-case alphabetic characters.

Focal Stream Ψ: The corpus restricted to the focal items.

Peripheral Items: The 35 most commonly occurring characters in the corpus.

Peripheral Stream: The corpus restricted to the peripheral items for $\mathbf{\Delta}_i(\Psi)$, for $i = \pm 1, \pm 2, \pm 3$.

Best Metric: Spearman, when the table is 'normalised'.

[2] Although there is evidence that it does not universally apply.

Cluster analysis was performed on the resulting contingency table, using a variety of metrics, the resulting dendrograms of which are shown in figure 6.5. As can be seen, Spearman Rank Correlation Coefficient gives a clear distinction between vowels and consonants. Interestingly, the $\mathbf{L}_1$ statistic, so good on the artificial data, does not perform so well on real data — that is, its results do not conform so closely to other theoretical classifications between vowels and consonants as the Spearman statistic.

This is best illustrated by figure 6.5, where the results of clustering the lower case characters from the same data are compared for various metrics. In all of the following, the data was first normalised, and then clustered.

6.2.5 Phonemic Analysis

A moderate sized corpus of phonemically transcribed speech (Svartvik & Quirk, 1980) was analysed using a similar contingency table statistic, the only difference being that this time the items were descriptions of phonemes. The phonemic alphabet used is the international machine-readable phonemic alphabet.

Corpus: 12,000 phonemes of transcribed speech (The Lund corpus, Svartvik & Quirk 1980).

Focal Items: 36 machine readable phonemes.

Focal Stream, Ψ: The stream of the corpus.

Peripheral Items: The same 36 phonemes.

Peripheral Streams: $\mathbf{\Delta}_i(\Psi)$, for $i = \pm 1, \pm 2, \pm 3$.

Best Metric: Spearman, when the table is 'normalised'.

The contingency table was thus,

$$\overline{[\Psi, \mathbf{\Delta}_1(\Psi) + \mathbf{\Delta}_2(\Psi) + \mathbf{\Delta}_{-1}(\Psi) + \mathbf{\Delta}_{-2}(\Psi)]}$$

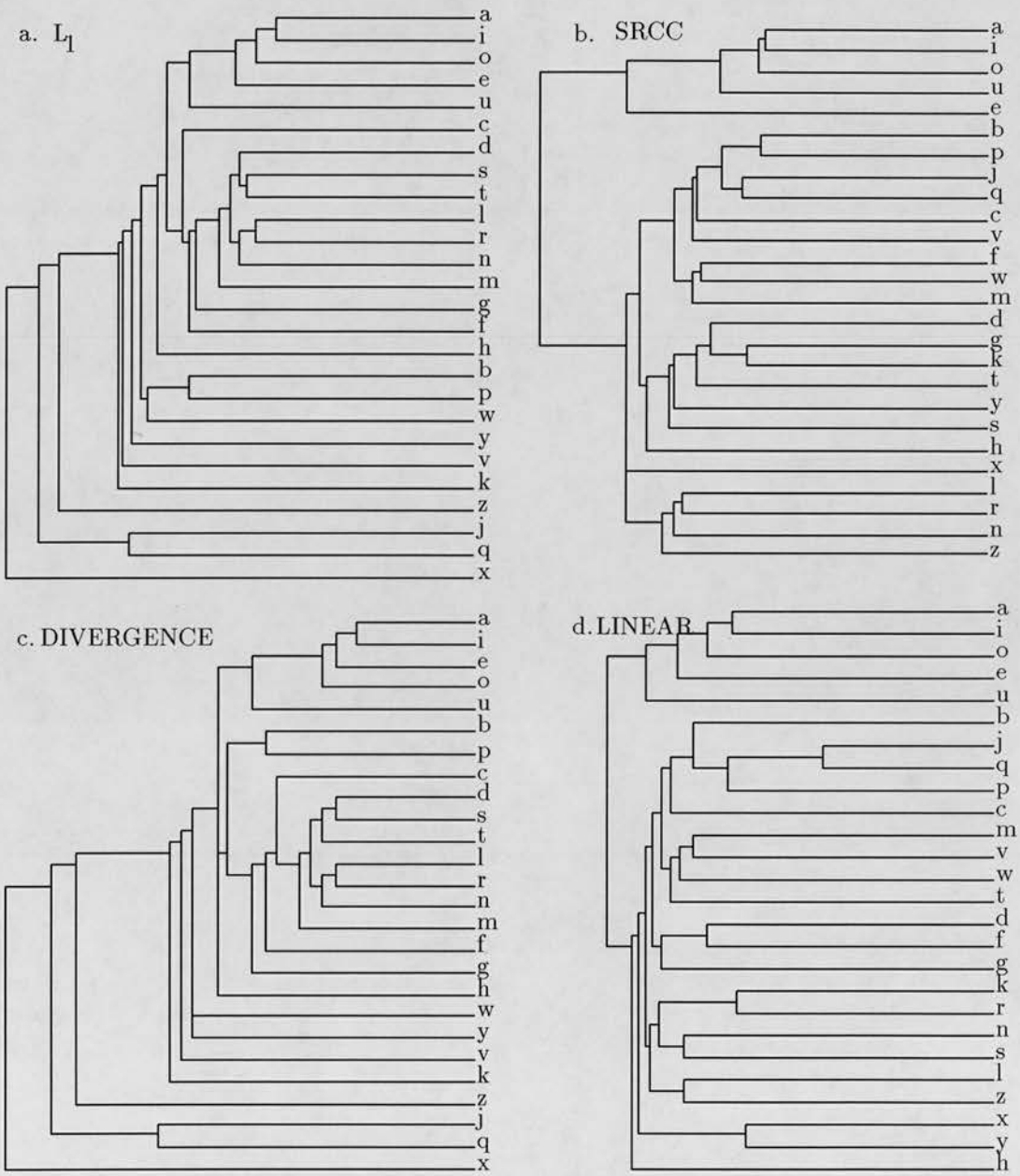


Figure 6.5: This figure shows how various metrics clustered the same data. As can be seen, part (b), corresponding to the Spearman Rank Correlation Coefficient, shows a clean cut between the vowels and the consonants. Part (d), corresponding to linear correlation coefficient shows a clean distinction too, but the distance between the vowels and the consonants is smaller, indicating a less robust distinction. The others all have extraneous 'outlyers' (a, c).

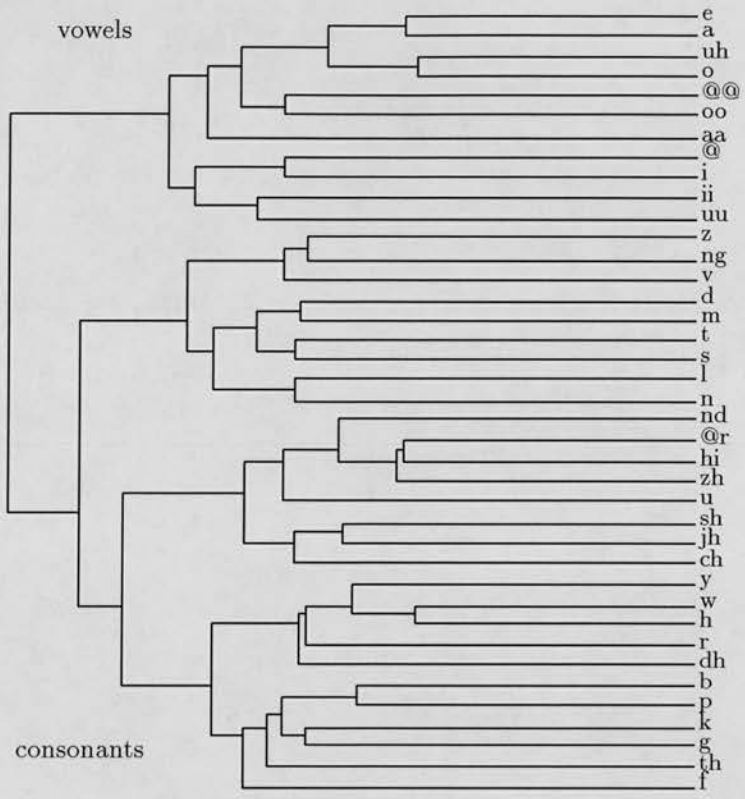


Figure 6.6: Dendrogram of phonemic data. The focal items are the machine-readable phonemic alphabet.

Again the Spearman rank correlation coefficient gives the dendrogram (figure 6.6) most in accord with intuition, L_1 misclassifying several vowels. Even the Spearman measure misclassifies one vowel, but this occurred only eight times in the whole corpus.

6.3 Words in Natural Language

The major experiment performed and reported here was on a large corpus of natural language. This section relates the results of some experiments in the classification of a natural language lexicon.

Corpus, Ψ: 45,000,000 words from USENET newsgroup articles over a four month period. The articles were initially parsed into 'phrases' so as to include to some extent the information accorded by punctuation. The definition of a 'phrase' is 'that which is between punctuation marks', which were defined as Comma (,;:())

and Period (.?!). Single hyphens were replaced by spaces. If a phrase was observed to contain an unknown character[3], it was discarded as being 'noisy'. Before the start of each phrase, the special item 'Start' was added, and after the end of the phrase, the appropriate punctuation item was added. Always, uppercase characters were converted to lower case.

Focal Items: The 2000 words found to be most common on one particular day in Newsgroup articles.

Focal Stream: The corpus restricted to the focal items.

Peripheral Items, P: The 147 most common words in the corpus together with three special items: Start, Comma, and Period.

Peripheral Streams: $\Delta_i(\Psi)$ for $i = \pm 1, \pm 2$, restricted to the peripheral items, P.

Contingency Table: $[\Psi, \Delta_{-2}(\Psi)_{|P} + \Delta_{-1}(\Psi)_{|P} + \Delta_1(\Psi)_{|P} + \Delta_2(\Psi)_{|P}]$

Best Measure: The measure found to be empirically best was the Spearman Rank Correlation Coefficient after normalisation.

The sum total of all cells of the contingency table was approximately 112 million (the sum of elements in each peripheral relation, therefore, would have been approximately 28 million).

Even a list of nearest neighbours shows a considerable amount of interesting linguistic structure has been captured by the metric.

three: three, four, two, five, six, white, black, red, security, commercial.

I: i, we, you, they, he, she, i'd, i'm, i've, i'll.

south: south, east, west, north, war, public, government, tv, system, dead.

the: the, our, my, his, your, their, its, a, every, another.

[3] Alphabetic characters were 'known', together with the punctuation marks mentioned above, and the apostrophe.

of: of, from, for, in, between, against, at, by, on, with.

year: year, day, night, group, game, line, crime, class, book, service.

are: are, were, aren't, is, was, isn't, wasn't, gets, they're, will.

can: can, could, cannot, can't, will, should, won't, might, would, couldn't.

remember: remember, understand, know, see, believe, think, read, consider, feel, mention.

small: small, large, short, big, simple, special, great, good, high, little.

These lists appear to show effects of both syntactic and semantic similarity. The first reveals a high correlation between "number" words. Numbers higher than six are not frequent enough to be considered in the data set, and "one" has a rather different distribution, since it has more than one grammatical function.

The subject position pronouns which cluster with 'I' do not include 'it', presumably because 'it' is an object pronoun as well. Words referring to time tend to cluster together. This is even more marked in the dendrogram, where temporally related nouns cluster together even though they are of differing syntactic category. Verbs which take a sentential complement such as 'know', 'wonder', and so on, tend to cluster together slightly away from other verbs.

The resultant dendrograms are very illuminating, giving a syntactic hierarchy well in accord with linguistic intuition. Figure 6.7 shows the overall structure of the dendrogram, while the other figures show local structure within it.

Looking at the dendrogram of artificial data generated from a context free grammar (figure 6.4), it is clear that the overall structure of the real data and the artificial data is similar. If we look at finer detail within the dendrogram, semantic as well as syntactic regularities become apparent. Figure 6.8 shows the dendrograms of the 'past-participle' node, the 'ing-form' node, and the 'auxiliary' node of figure 6.7. It is clear that words of similar meaning tend to occur in similar contexts. Examples include said/ thought/ felt/ decided/ .../wanted (words to do with thought); posted/ mentioned/ released/

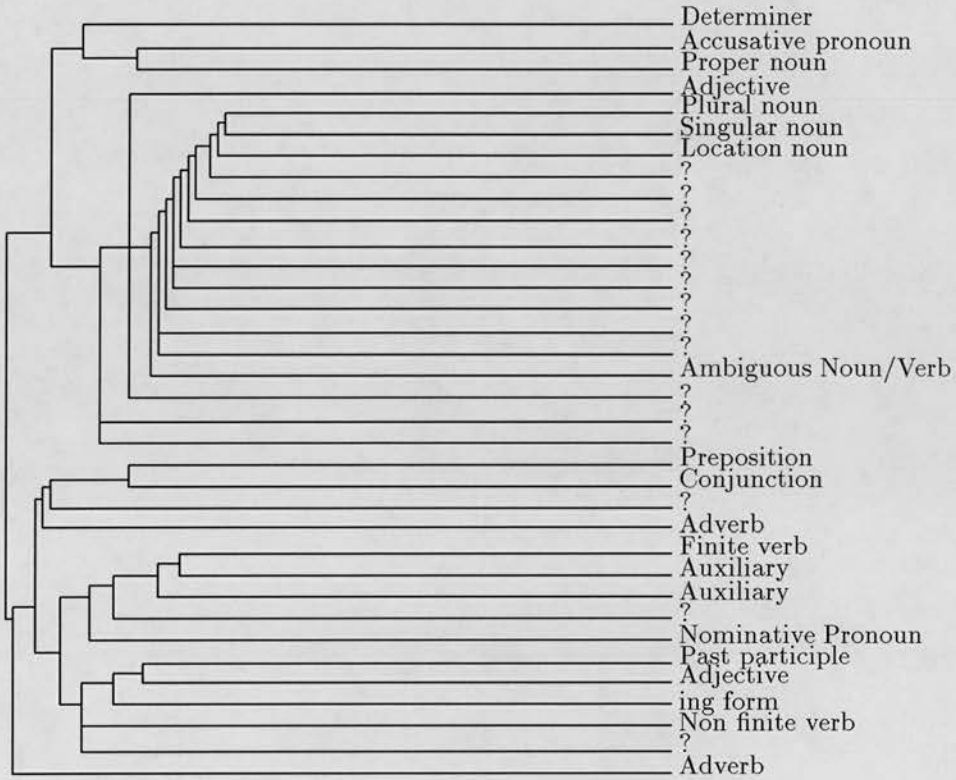


Figure 6.7: This figure shows the overall structure of the dendrogram of lexical items. Parts of the structure labelled '?' correspond to small branches of lexical items with fewer than 20 occurrences. The total proportion of the data judged not to fall within this overall classification is less than 5%. The labels given here are standard linguistic labels which were deemed to most accurately describe the contents of the cluster.

.../published (words connected with the provision of information); going/ coming/ looking .../ fighting/ acting (words concerned with movement and action), and so on. It is interesting to note that these regularities are apparent with context being differentiated only on the 147 most common words. Few of these words are content words, so intuitively useful linguistic regularities such as subcategorisation frames will not be available.

Figure 6.9 shows the preposition node, and the nominative and accusative pronoun sub-nodes of figure 6.7. It is clear that very considerable syntactic structure has been extracted by representing the similarity metric in this dendrogram. In particular, there is a strong division between pronouns which refer to people (he, him, she, me, I, someone, anyone etc.), and pronouns which refer to inanimate objects (it, something, this, anything, etc.)

Within the noun categories, very many considerable low level semantic regularities are apparent to casual inspection. Figure 6.10 shows several subtrees from within the singular noun node of figure 6.7. It is evident that considerable semantic structure is evident in these dendrograms. The upper-left dendrogram has a preponderance of words used to discuss computers, while the lower left hand dendrogram has a preponderance of words used to refer to the units of society. The upper left hand box has nouns which refer to people, while the lower right hand box contains predominantly mass-nouns.

6.3.1 Lexical Ambiguity

On theoretical point which must be mentioned here is that of lexical ambiguity. A formal linguistic analysis of a corpus of natural language will not always assign a word the same category for all its occurrences. This gives rise to *lexical ambiguity*, which this classification procedure is unable to directly account for. Practically, the problem is that some words can fill a number of different linguistic roles. For instance, the word 'cut' can be a noun ('a nasty cut'), an adjective ('loaves of cut and uncut bread'), an infinitive verb ('Mary told John to cut and eat the cake'), a past tense verb ('John cut and ate the cake'), or a past participle ('After John had cut and eaten the cake, Mary thanked him.'). However, *cut* must be assigned a single position within the hierarchy,

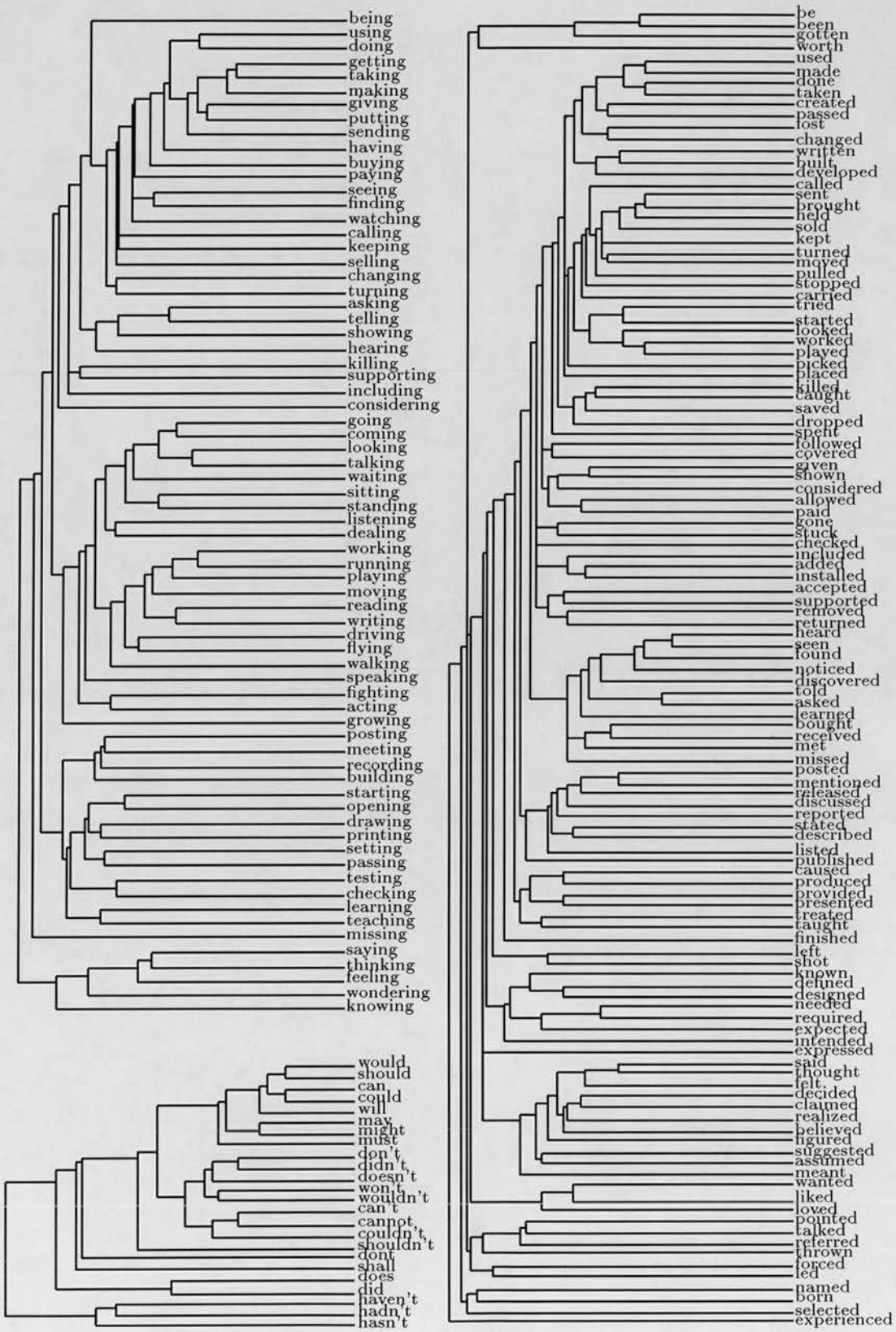


Figure 6.8: Detailed structure from within the past-participle and ing-form nodes of the overall

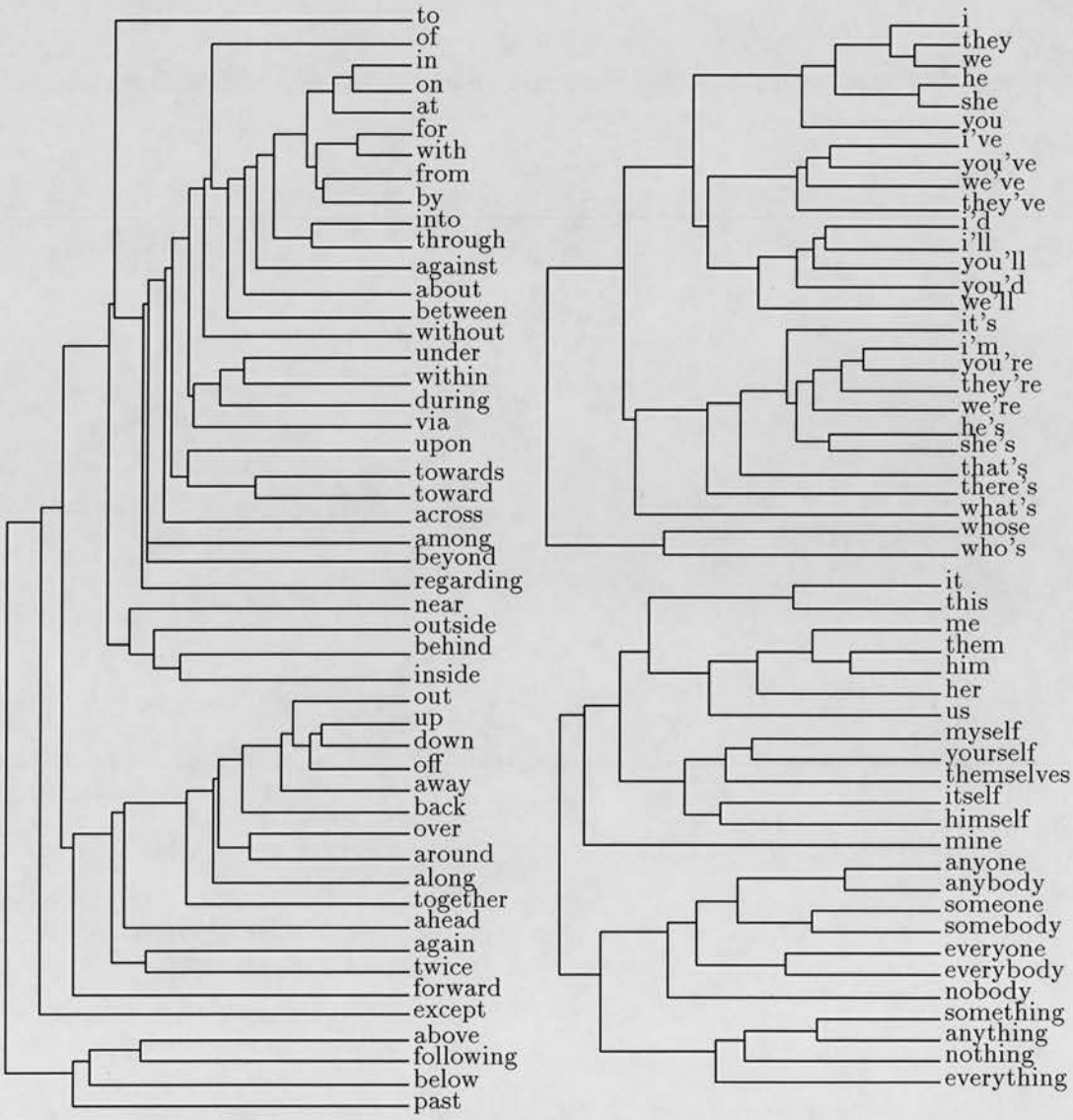


Figure 6.9: This shows the internal structure of the preposition, and the accusative and nominative pronoun nodes of the overall dendrogram.

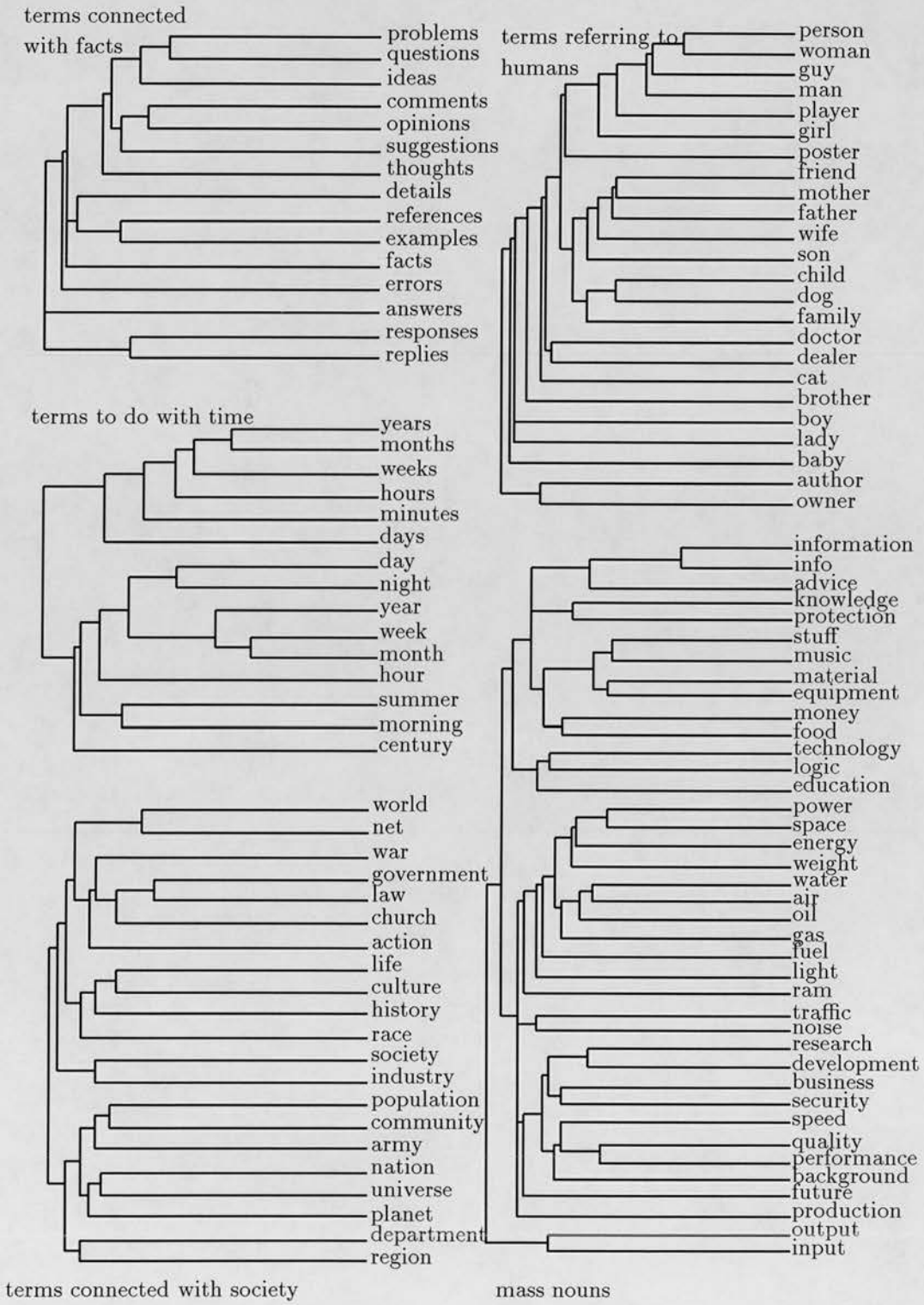


Figure 6.10: Some of the dendrograms within the noun node of the overall picture show some more syntactic and semantic structure.

and this does not fit well with linguistic interpretation of 'class'.

An analysis of the dendrogram, however, shows that words which are often assigned different categories form a subcategory of their own. This is especially true of words which are sometime nouns, and sometimes verbs. Figure 6.11 shows one such cluster of words which contains words which are often assigned more than one category.

6.3.2 Empirical Evaluation

Recall that the process of clustering described in section 5.4 involved iteratively combining clusters to form a binary branching tree of clusters. It follows that it is possible to stop this process at any stage, resulting in N clusters which represents the current classification of the lexicon, where N is anything between 1 (the whole lexicon) and the number of individual lexical items (equivalent to doing no clustering at all). For all N, the N classes which exist form a partition of the lexicon — that is each lexical item appears in precisely one of the N classes. An equivalent way of visualising this is by generating the entire dendrogram and 'cutting' it at a certain dissimilarity level. The nodes directly above the branches which have been cut are the roots of sub-dendrograms, the leaves of which form the partition.

For natural language, if the clustering process is halted when 400 categories remain, a partition is derived. The 100 categories at this level which comprise the greatest part of tokens in the corpus is given in appendix C. Our attention is restricted to these 100 categories because many of the remaining 300 categories consisted of single, infrequent words which would have aided the classification procedure very little, while greatly expanding the number of categories[4]. On the other hand, if clustering had continued until only 100 categories remained, many of the resulting categories would be linguistically incoherent. Again, as was the case in the decision to restrict the peripheral items to the 150 most common word types, the decisions about when to stop clustering,

[4]The practical reason for discarding the other 300 categories is that the classification described here is the classification used as the basis for the analysis of word sequences described in the next chapter. There, sequences of categories of length 3 are analysed, and the number of such sequences is cubic in the number of categories, so restricting the number of categories by a factor of four, reduces time and space complexity in collecting and analysing these sequences by a factor of 64. Since this was the classification used there, this is the classification empirically evaluated.

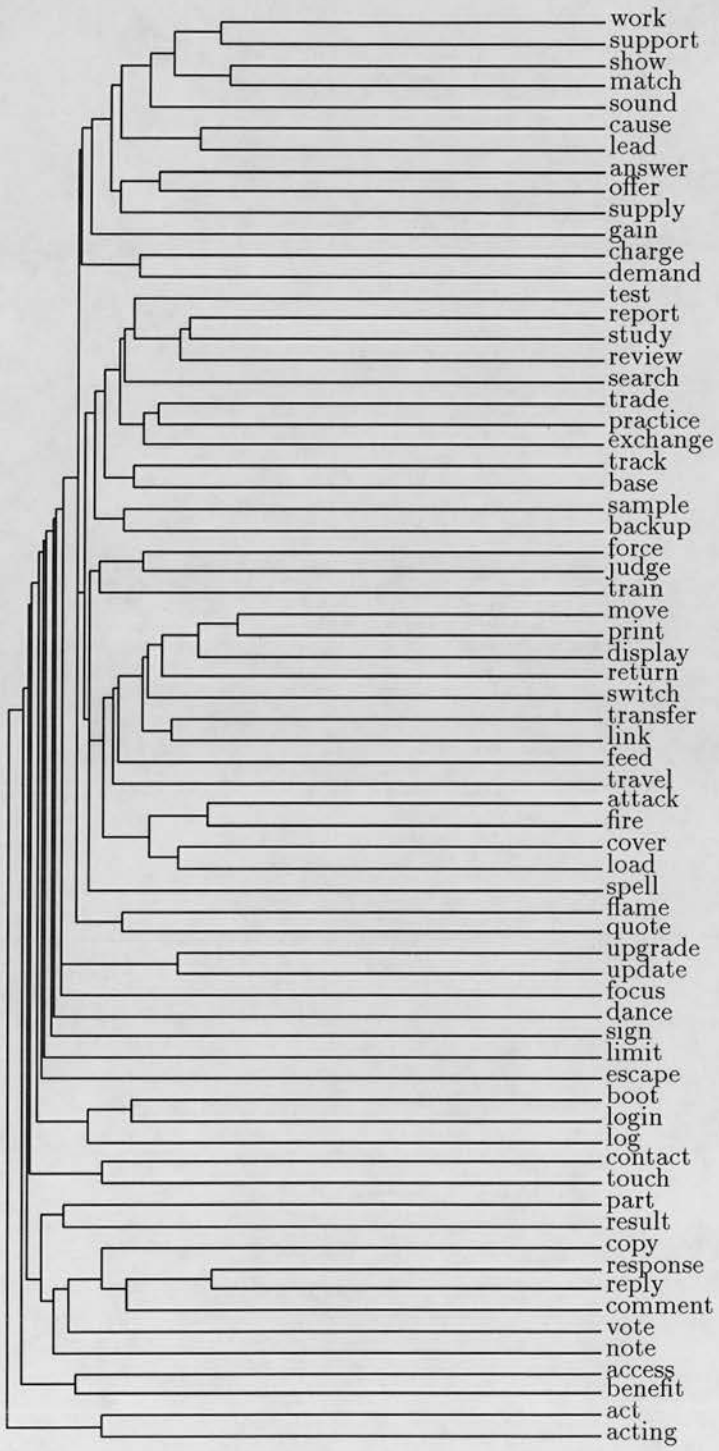


Figure 6.11: A dendrogram of a subcluster of words which are frequently assigned more than one tag in a formal linguistic analysis of the contexts in which they appear.

and about how many of the remaining clusters to retain, were arbitrary.

This partition includes 75% of the words in the original 2000 words which were surveyed, and over 90% of word tokens which were tokens of the original 2000 words occurring in the corpus appear in these 100 categories (since many of the types omitted were relatively infrequent). This list, therefore, defines a functional classification of words into categories. In what follows, C_n refers to that set of words in the nth category listed in appendix C.

The SUSANNE corpus (Sampson 1992) is a subset of the BROWN corpus which has been tagged and parsed by hand. The words were tagged with the Lancaster tag set (Garside et al. 1987), which is a relatively fine-grained tag set comprising some 352 types of word tag[5]. In this tag set, distinctions too fine grained to be captured by a coarse ontology, such as that derived here, are derived. For instance, the words *the*, *his*, and *their* are all assigned different tags in this tagset for their most common use as a determiner of a noun phrase. In order to empirically evaluate the efficacy of this classification procedure, the following two classification schemes will be compared:

1. The *reference Lancaster tagger* is defined to be a mapping between words and Lancaster tags such that a word $w \in W$, for some given set of words W, receives the Lancaster tag most frequently associated with it in some tagged training corpus.

   The performance of the reference Lancaster tagger on a tagged corpus X is defined to be the proportion of word tokens of types in W which it correctly tags.

   Although there are over 350 possible Lancaster tags, only 40 of these were the most common tag for any word of the 1450 word types which appear in appendix C. Thus the reference tagger can be thought of as a mapping of word types into these 40 tag types.

2. Use the empirically derived classification scheme to tag each word. Calculate the mapping, F, between the empirical tags and the Lancaster tags such that $F(C_n)$ is the Lancaster tag most frequently associated with the word tokens whose types appear in C_n.

[5]Although there are 352 tag types, many of these are very rare. Over 95% of tag tokens in the SUSANNE corpus are accounted for by 50 tag types.

The measure of empirical validity is the proportion of word tokens which received their correct Lancaster tag under this mapping. This tagger will be referred to as the *empirical Lancaster tagger.*

The ratio of these two numbers will be called the *efficiency* of the empirical classification. The efficiency of the empirical classification derived here for the classification described above and listed in appendix C, is about 70%. That means that 70% as many words receive their correct Lancaster tag if the classification is done on the basis of the empirical categories of the words as described above, than if the classification is done with full knowledge of the identity of the word according to the reference tagger. By contrast, a random mapping of word types into categories, which retains the same number of types in each category (e.g. C100 had 4 random word types in it) had an efficiency of about 9%. Consequently, as one would expect from a casual analysis of the empirically derived categories, the empirical word classes are highly correlated with linguistically motivated classes.

Under this evaluation measure, it is possible to get an efficiency of more than 100%. This happens, for instance, in the following (unlikely) situation. In the training corpus for the reference Lancaster tagger, a word w, in empirical category C_n, most frequently receives tag L_1, but the word tokens in C_n most frequently receive tag L_2, so the empirical Lancaster tagger maps w to L_2. The test corpus is a sequence of ws tagged with L_2, $w/L_2 w/L_2 w/L_2$.... Then the direct "word to Lancaster tag" tagger scores 0%, while the "word to empirical tag to Lancaster tag" tagger scores 100%, making the efficiency either undefined or infinite.

However, real language is not like this, and if the test corpus of the reference tagger were to be the same as the corpus on which it was trained, then this scenario is not possible, and the maximum efficiency of 100% is achieved when the empirical categories are defined so that each empirical category contains only words which map to the same category in the reference Lancaster tagger. The reference tagger can be thought of as an empirical tagger with a trivial empirical classification (each word maps to its own distinct category). Viewing it in this way, it is clear that we would not normally expect the empirical tagger to do so well as the reference tagger, provided that the test corpus

is statistically similar to the training corpus, so the efficiency will normally be less than 100%.

One might ask why there is only a 70% efficiency rate, given that it was claimed that 95% of word types the dendrogram in 6.7 was linguistically coherent? The answer is that there is not a single accepted definition of "linguistically coherent". The major sources of disparity between the reference tagger and the empirical tagger were due to the Lancaster tagset having different tags for different prepositions and determiners. For instance, the tag for "the" is "AT", while the tag for "a" is "AT1". Both these words would receive the generic linguistic description of "determiner" in their usual usage, as would possessive personal pronouns, such as "his", "their", and so on. Indeed all these words are in the same empirical category (C_1), so this Lancaster distinction is not present in the empirical categories. Also, different prepositions receive different tags (there are around 10 different tags associated with different uses of prepositions), and there are many different verb tags. None of these distinctions are reliably present in the empirical categories.

The other widely used tagset is the BROWN tagset. This embodies a much broader classification, with many fewer tags (approximately 40 word level tags). The Data Collection Initiative of the Association of Computational Linguists has collected a corpus of Library of America texts, Wall Street Journal articles, and U.S. Department of Energy abstracts, semi-automatically tagged with the BROWN tagset to an accuracy of within 2% (Liberman et al, 1991). If the same procedure is done with this corpus as was done with the SUSANNE corpus above, with the reference tagger being trained on Wall Street Journal texts, and tested in Library of America texts[6], the efficiency of the empirical classification with respect to this tagset is 92%, with 169 words receiving a different most common BROWN tag than the most common tag of their empirical category. The major source of disparity between the empirical tags and the BROWN tagset is again the fact that some empirical determiners[7] are tagged as possessive personal pronouns, and the distinction between "VBP: verb, non-3rd person singular present" and "VB:

[6]It was thought this was a fairer test, since the empirical categories had been derived from USENET articles, not Library of America texts. The mapping between empirical categories and BROWN tags used Library of America texts.

[7]i.e. Some words in the same empirical category as "the" (C_1).

Verb, base form", is not accurately captured by the empirical categorisation.

6.4 Statistical Reliability

How many instances of a word are needed before the methods described here can reliably incorporate it into a taxonomy? The taxonomy is derived from a word-word distance table, so an associated question is "how confident are we about the measures of distance between two words?". The correct way to answer this question is to find an expression for the variance of the distance measure $d(w_1, w_2)$ under the assumption that w_1 has n_1 entries in the contingency table, and w_2 has n_2 entries in the contingency table. We then define some arbitrary number c, and say that $d(w_1, w_2)$ can be reliably estimated provided that $Var(d(w_1, w_2)) < c$.

Unfortunately, such a calculation depends on the distance metric d used, and on what we would expect a-priori the distribution of the row of the contingency table associated with w_1 and w_2 to be. We might assume that the frequency of bigrams starting with w_1 satisfies Zipf's law, but since we are only considering bigrams involving the 150 most common words, it does not follow that the distribution over these words is well approximated by Zipf's law.[8]

This means that it is very difficult to make assumptions about the distribution of the rows of the contingency table which are both realistic and mathematically tractable. However a good rule of thumb has been found to be that an estimate may be considered reliable if the sum of the row contingency table is at least twice the number of cells.

However, there is a method which allows us to ignore problems with reliability of distance estimates when constructing hierarchical clusters. The method involves adding a real random number between 0 and c to each cell of the contingency table. This will have an insignificant effect on the estimated distances between words which have a large number of observations for small c (typically less than 1), but will have a significant effect where

[8] Indeed, when one looks at the distributions, one finds a larger than expected number of zero, or very small, numbers. This is because some words in the top 150 words are restricted from occurring next to the focal word for primarily syntactic reasons. For instance, for the focal word *his*, there are very few following verbs because in its most common use as a determiner, *his* does not permit a verb to follow it immediately.

the number of observations is small. Consider, for instance, the metric L_1. Consider the distance between w and w', where w' is w with noise added to its vector as described above with $c = 0.5$. Let us write w_i for the ith component of the vector associated with w, and similarly $w'_i = w_i + r_i$, where r_i is a vector of random numbers uniformly distributed between 0 and 1. Consider $d(w, w')$ for the case where $\sum_i w_i = N$.

$$d(w, w') = \sum_{i=1}^{600} |\frac{w_i}{N} - \frac{w_i + r_i}{N + \sum_j r_j}|$$

Which, after some rewriting, becomes

$$(6.1) \quad d(w, w') = \frac{1}{N + \sum_i r_i} \sum_i |\frac{\sum_j r_j}{N} w_i - r_i|$$

So, taking term by term inequalities over the summation in 6.1, we see that $d(w, w')$ is bounded by

$$(6.2) \quad \frac{1}{N + \sum_i r_i} \sum_i \frac{\sum_j r_j}{N} w_i - r_i \leq d(w, w') \leq \frac{1}{N + \sum_i r_i} \sum_i \frac{\sum_j r_j}{N} w_i + r_i$$

Simple algebra reduces 6.2 to:

$$0 \leq d(w, w') \leq 2 \frac{\sum_i r_i}{N + \sum_i r_i}$$

which decreases with N. For example if the expected value of $\sum_i r_i$ is 150, as is the case if noise in $[0, 0.5]$ is added to each cell, and $\sum_i w_i = 1200$, then $d(w, w') < \frac{2}{9} \approx 0.22$. This is an upper bound, and the expected value of $d(w, w')$ depends on the distribution of the $\{w_i\}$, but in random tests was found to be approximately 0.07.

This shows that, at least for the L_1 metric, adding random noise to a vector has an effect on the distance between the original vector and the noisy vector such that the upper bound of the distance decreases inversely with the total number of tokens of the word in the corpus. Thus this technique can be used to "push apart" infrequent word types so they do not interfere with the hierarchical structure derived from the distance metric between words without pushing apart frequent words for which we have reliable statistics.

This technique was used on all the experiments on natural language reported in this thesis.

6.5 Semantic clustering

Brown et al. (1990) report a set of similar experiments to the ones described here in which a taxonomy of words was obtained similar to the taxonomy obtained above, and in which a semantic taxonomy of words was obtained. The means by which this was achieved is similar to the methods described here, except the goal, rather than to find a metric between words and cluster on the basis of this, was to find that classification which yielded a 'best' class-based bigram model.

Although the details of their approach are quite complicated, roughly they defined a space of 'class-based bigram models' for a fixed number of classes. A 'class-based bigram model' is a very simple hidden Markov model (see chapter 3), where there is a many to one mapping between lexical items and states[9]. It is possible to measure how well a particular HMM models a corpus by dividing the number of words in the corpus by the log of the probability of the corpus. This yields a measure called *perplexity* — the lower the perplexity, the higher the prior probability of the corpus, the better the model.

Brown et al. defined an algorithm which would find a model at a local optimum of perplexity, in the sense that if any single word was placed in another class, the resulting model would have higher perplexity than the locally optimal one. The resulting classification was extracted. This was also the approach adopted by Kneser & Ney (1991). It should be noted that this approach does not take account of information concerning dependencies between non-adjacent words, as the technique described above does, but on the other hand it provides a sound theoretical link between the notions of classification and prediction, which is the dichotomy underlying the bootstrapping problem.

In order to obtain a semantic taxonomy of words, the notion of 'stickiness' is introduced. A pair of words $\langle \psi, \phi \rangle$ is *sticky* according to a relation, R, if the pair occurs in the relation

[9] i.e. each word is associated with only one state. This means that the set of states can be thought of as partitioning the set of words into disjoint categories. This means that the sequence of state transitions is determined by the sequence of words, so it is a very much simpler class than general HMMs.

more often than one would expect from knowing only how often each word appeared in the relation (i.e. how often ψ occurred in the first argument, and ϕ occurred in the second position of R). For instance, Brown et al. reported that if the relation is 'next word', then pairs like *Humpty Dumpty*, *mutatis mutandis*, and *avant garde* were the stickiest pairs. If the relation R is defined as 'occurs within 500 words of the current word, but not within 2 words' then Brown et al. found that if they clustered according to the stickiness of words in this relation they got clusters with a common stem such as *performance, performed, perform, performs, performing*, and clusters of semantically related words such as *counsel, trial, court, judge.*

In order to find semantic regularity (such as semantic word associations), it is necessary to find a statistical relationship whose regularity is due mainly to semantic factors. One such, as Brown showed, is lexical co-occurrence within a certain window of words, and the semantic regularity exists because people usually write about the same subject for several hundred words before moving on to another subject.

Another, non-linguistic, possibility will now be explored. Associated with every news article in the corpus is a news group which contains (largely) articles discussing a certain subject area. The newsgroup of an article is readily available, and consists of a number of fields (e.g. sci.math.num-analysis contains the fields 'sci', 'math', and 'num-analysis'.) The statistical regularity we shall be interested in is the regularity between the words which appear in an article, and the fields of the newsgroup in which the article appears.

6.5.1 Experiment in Deriving Semantic Structure.

In this section, each article is represented by the multi-set of its words, and each newsgroup by the multi-set of its fields. We are therefore co-indexing two multi-streams (see definition 5.1.6), and taking their contingency table according to definition 5.1.8.

Suppose we have two co-indexed streams, Ψ_W and Ψ_N, Ψ_W being a stream of multi-sets of words, and Ψ_N being a stream of multi-sets of fields of newsgroup names such that the multi-set $\Psi_W(n)$ is the multi-set of words in an article which was observed to occur in the newsgroup with name which had field names in $\Psi_N(n)$. This stream,

then, is our corpus. The contingency table of words against newsgroup field names was collected, and analysed exactly as the contingency tables used for uncovering syntax were analysed.

In order to present the results of the classification, the dendrogram was generated and then cut at a low level. Any clusters which were formed below this level of similarity which contained more than one word are printed below.

6.5.2 Results

1. you your
2. i my me know
3. was had they were said did
4. he his him
5. but just don't it's really so like that's didn't
6. out off up back
7. go going
8. one about very good
9. more than much most
10. when after
11. we our
12. use using uses
13. file files directory code unix user users
14. ftp anonymous
15. system systems
16. software interface application applications data

17. program programs
18. machine machines
19. dos pc ibm ms
20. install installed installing installation configuration setup
21. patch patches
22. scsi ethernet lan
23. modem baud modems
24. manual manuals
25. upgrade compatible
26. buffer byte bytes interrupt reset
27. cache apps app vendors vendor developers
28. utility utilities
29. vga bios ide borland
30. server servers protocol client clients
31. nfs bsd ultrix perl sco sparc
32. int char gcc struct unsigned init ptr filename
33. compiler compilers ansi fortran routines parameter integer pointer pointers
34. error errors
35. menu cursor icon
36. supports supported
37. connected connect
38. device devices

39. chip chips

40. mount mounted

41. faster slower

42. story knew

43. live living

44. woman girl boy girls boys she's loved lady kid

45. married feelings

46. watching watched caught

47. wear wearing hair

48. blood skin

49. law laws

50. jews jewish catholic christians jesus

51. religious religion church

52. sex sexual gay

53. population nation nations citizens economic civil

54. violence violent innocent victims murder armed justice crime crimes criminal

55. democratic democracy politics republican soldiers troops communist minister

56. fight fighting attacks attacked threat defend

57. argue arguing

58. song songs

59. alien planet

60. votes voting

61. armenian armenians ottoman kurds turks villages armenia turkish muslim muslims islam islamic arab genocide iran

62. went took

63. buy bought buying

64. conference papers

It is clear that the semantic structure uncovered by this procedure is not so impressive as the syntactic structure derived above. Nevertheless, this is another example of how the non-deterministic regularity based classification technique may be applied to uncover structure in natural language.

6.6 Conclusion

As a technique for the unsupervised recovery of categorical information, the technique presented here seems to work well. In both artificial and real data sets, considerable structure has been shown to be uncovered by simple statistical analysis of redundancy. In artificial data, a classification consonant with the generating process was shown to be derivable both for simple two and three word sentences, and for a more complicated stochastic context free grammar. For letters, a strong vowel/consonant distinction was shown to be derivable from real data, and for phonemes, the same distinction was observed. For natural language, all of the traditional syntactic categories were evident from the dendrogram clustering newsgroup data, and within these syntactic categories, considerable semantic structure is observable.

Thus, by representing lexical items in a way which expresses the redundancy due to the process which generates it, unsupervised clustering can uncover much structure, at least at the lexical level. There are two ways to continue the exploration of this approach: Firstly, we can look for other, possibly better, clustering mechanisms to find the regularities which have already been uncovered. These techniques may be compared and contrasted, and conclusions as to which is the best technique may be drawn. In chapter 8, I consider some algorithms, from the literature on neural networks, for forming

topographic mappings, and apply these to the natural language data. Secondly, we may look to uncover 'higher-level' structure from natural language. Linguists typically split sentences into structural units, or *phrases*. These phrases can be used to explain why some sequences of words are *ill-formed*, and they can be used to analyse what the sentences mean. Phrases are defined over sequences of words, and whether generalisations of the techniques already introduced here can uncover phrasal structure is an interesting question which will be addressed in the next chapter.

Finally, it is necessary to mention that although the classifications which result have been discussed, and it is clear that they *appear* to correspond to orthodox classifications of the domain in question, no empirical criterion for a 'good classification' has been proposed. Why should the orthodox classifications be the ones to be aimed for? One reason is that they are orthodox precisely because these classifications have been shown to be useful in expressing regularities and generalisations over language. As mentioned in chapter 2, classification is useful only insomuch as it supports inference and generalisation. One might then ask the closely related question: If a good classification has been uncovered, how is it possible then to find a good theory which actually expresses the redundancy in the data?

This, of course, is another way of presenting the general learning paradigm presented in chapter 4, and all the complexities of the argument presented there apply. However, a small step towards an answer to this question can be given if statistical models of the natural language stream are made, and if the classification of language derived here can be shown to be useful in creating such models.

Chapter 7

Analysis of Sequences of Words

In line with the analysis of theories of language presented in chapter 3 after Halliday, we turn our attention now to higher elements of structure in natural language. We have already seen that unsupervised techniques can be relatively successful in uncovering word classes. This chapter shows that the same techniques can be relatively successful at uncovering phrasal classes — classes of sequences of words which are likely to play the same role in linguistic analyses of sentences. The first part of the chapter reviews the linguistic approach to structure, largely after Chomsky (1957), but with reference to a Hallidaean framework. The rest of the chapter details two experiments in uncovering phrasal linguistic structure within the unsupervised statistical paradigm of this thesis.

The work described in this section has some relevance to the framework of hierarchical language analysis described in Powers & Daelemans (1992), and the experiments performed by Powers (1992) (with somewhat mixed results) on the derivation of a hierarchical orthographic ontology from natural language texts, where longer and longer sequences are distributionally analysed and classified.

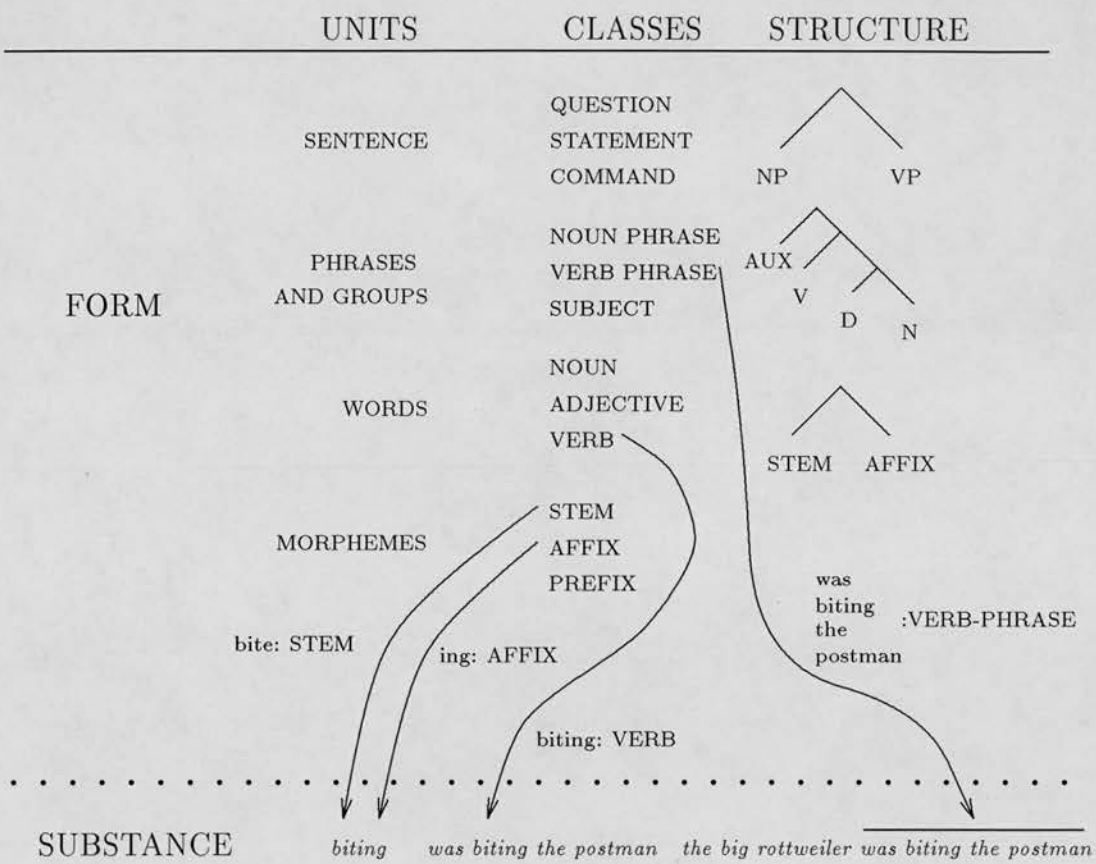


Figure 7.1: This figure shows the relationship between the various components in Halliday's analysis of linguistic theory. In particular, it shows how the substantial sentence *the big rottweiler was biting the postman* might be broken up in a standard linguistic analysis.

7.1 Words, Phrases and Linguistic Theory

According to Halliday, the *units*[1] of linguistic theory form a linear hierarchy. The smallest unit is the *morpheme*. Morphemes are grouped into structures according to the rules of morphology to form *words*, and these are grouped according to the rules of syntax to form *groups* or *constituents*, some of which are *phrases*. The rules of syntax also group constituents into syntactic structures to form the highest linguistic unit — the sentence.

In *form*, the structure of clauses and phrases is theory (system) dependent, but typically the structures assigned are labelled tree structures. Figure 7.1 shows the relationship

[1] See section 3.1.3

between the elements of Halliday's analysis of the grammar of language. The relationship between *structure* and *class* is of particular interest to this thesis. Structures are defined in terms of classes of the unit below. Thus phrases contain word classes as terms, and word classes in turn are defined as features of words in context which give rise to linguistically perspicuous[2] structural descriptions.

As was discussed in section 3.1.3, the definitions of class and structure are not independent; the one cannot be thought of as logically preceding the other. The distributional evidence, such as that presented in section 3.1 shows that the mapping between words and classes should not be regarded as *functional.* A word does not always play the same linguistic role in every structure in which it features; sometimes it might be interpreted as a noun, other times as an adjective, and yet other times as a verb. Which interpretation is assigned to a particular (substantial) token of the word is dependent on the interpretations of the structure of the unit in which it features, and this, in turn, is decided by the linguistic role it plays in the phrase or sentence in which it features.

7.1.1 Formal and Substantial Structure

There is evidently a difference between the structures linguists postulate for words and phrases, which are typically labelled trees, and the substantial structures available, which are typically sequences of letters (in text), or sequences of phonemes (in spoken language). In the case of spoken language, the situation is further clouded by uncertainty about the nature of the representation of the sound stream. To be sure, it *can* be represented as a simple sequence, such as the sequence of magnetic variations on a tape recording of a speech act, but it is also possible to give this representation more structure, and represent speech as parallel streams of data, roughly corresponding to the positions the various parts of the mouth 'aim for' while making the speech sound. Often the speech act is represented as a sequence of *phonemes*, each phoneme corresponding to a distinct sound (or mutually disjoint classes of sounds).

[2]Linguistic perspicuity is the 'art' of linguistics — It refers to those structural descriptions of sentences which support certain generalisations (to other linguistic examples), account for various intuitions linguists have about the structure of language, and have a fairly transparent relationship with the meaning of the sentence (however 'meaning' is construed).

Whatever the nature of the representation of the language *substance* being investigated, for linguistic enquiry to proceed, it is necessary to propose a mapping between substantial structure (which exists in raw data, for example sequences of words) and formal structure (linguistically perspicuous representations). Already we have seen how to retrieve some classifications correlated with linguistically perspicuous classifications of words using unsupervised methods. These classifications are part of (or or correlated with) formal classes as described by Halliday. The next stage is to use these classifications to uncover further formal structure from observed substantial structure.

7.2 Linguistic Analysis of the Phrase

A sentence is typically analysed linguistically as comprising many overlapping constituents. Constituents are sequences of words which can be interpreted as a linguistic unit within a particular system in the Hallidaean sense.

For instance, the sentence

(7.1) The very fat cat ate the mouse.

traditionally is interpreted as having the following constituents beyond lexical categories such as verb, noun, adjective, and the like.

very fat — adjectival phrase;

very fat cat — n-bar;

the very fat cat — noun phrase;

the mouse — noun phrase;

ate the mouse — verb phrase;

the very fat cat ate the mouse — sentence.

In addition, some of these constituents have a special role in linguistic theories of syntax, and these are called *phrases.* Above, the phrases are *very fat*, *the very fat cat*, *ate the mouse*, and *the very fat cat ate the mouse.*

There are many empirical tests which can provide evidence about whether a given sequence of words in a sentence is a constituent of the sentence or not, and if so, what type of constituent it is. Some such tests are now described.

The first, and possibly most important, empirical test is the replacement criterion. Recall that this was the linguistic basis of the experiments in lexical classification described in the last chapter. It states:

> Can the phrase in question be replaced by another phrase of a known type?
> If so, then it is a phrase of that type.

Indeed, the replacement criterion will also be the basis for the derivation of classes of sequences of words. However, there are many other linguistic tests which can be applied, and these form the possible basis of other tests which might be applied. The first of the other criteria is the *movement criterion.* It states

> Can a sentence be formed which is closely related to the original sentence,
> but in which the phrase in question might be interpreted as having moved?
> If so, then the phrase in question is likely to be a constituent.

examples of the use of this criterion is given by linguistic constructions known as *preposal* or *postposal.* For example, consider the following pair of sentences.

(7.2) I can't stand her elder sister.

(7.3) Her elder sister, I can't stand (but she's OK herself).

In this example, the phrase *her elder sister* has been preposed to form a new sentence. In a sense, it has moved, and the movement criterion says that this can be used as evidence that *her elder sister* is a constituent.

Another criterion for constituent identification is the fragments of sentences which people give as answers to questions. for instance,

(7.4) Q: Where did you go last week.

(7.5) A1: I went to London.

(7.6) A2: To London.

(7.7) A3: London.

All three answers are possible, and all correspond to phrasal constituents in the standard analysis of *I went to London.*

Another criterion is that of *ordinary coordination*, which can be used to identify the type of a constituent once it has been identified as such. Phrases of the same type can be coordinated by conjunctions. For instance, consider the following sentence.

(7.8) The fat cat ate the mouse and the hamster.

The phrase *the mouse and the hamster* has two constituents — *the mouse* being one, and *the hamster* being the other. The coordination criterion asserts that these must be of the same type. It says:

> If a phrase can undergo ordinary coordination with a phrase of a known type, it is a phrase of that type.

There are other distributional tests which lend further weight to traditional syntactic analysis, but this gives a flavour of the empirical tests which have been developed, and which can be used to justify particular analyses for example sentences. However, these tests should not be thought of as *defining* constituency. For instance, according to all these tests *Subject-Verb* would be a constituent. For example, "Tom ate" can be replaced by "The fat cat had eaten"; it can undergo postposal as in "*That* mouse, Tom ate. (But *this* mouse, Sylvester ate)"; It can undergo ordinary coordination, as in "Sylvester killed and Tom ate that pesky little good for nothing mouse". Consequently, these tests can be used to support the utility of a general notion of constituency, rather than a particular definition of constituency. Indeed, the paradigm of "flexible constituency" (e.g. Barry & Pickering 1989) takes an alternative view of constituency derived from a notion of dependency in categorial grammars which interprets subject-verb units as constituents.

7.2.1 Immediate Constituent Analysis and Transformations

In analysing the structure of sentences in natural language, it is clear that some sentences have a close syntactic relation to each other. For instance, consider the following sentences below

(7.9) The very fat cat eats the mouse.

(7.10) The cat eats the mouse.

(7.11) Tom eats Jerry.

It is clear that (7.11) is the shortest sentence, that (7.10) can be derived from it by replacing *Tom* with *the cat*, and *Jerry* with *the mouse*. (7.9) can be derived from this by replacing *cat* with *very fat cat*. This intuition about 'replacing' words in a simple sentence to form a more complicated one was first embodied in linguistic theory as *immediate constituent analysis*. The intuition is, that to show that a certain string, s, is a sentence, all that is necessary to do is to show how it may be derived from a simple sentence, s', by repeated substitution from a simple sentence. Moreover, this analysis can also show what internal structure the sentence has, since structure can be defined to follow the derivation. In the above example, the structure justified by the substitutions suggested might be the following:

(7.12) [The [very fat cat]] eats [the mouse].

where each bracket pair derives from some substitution. Further linguistic analysis might provide a fuller specification of the structure as something like:

(7.13) [The [[very fat] cat]][eats [the mouse]]

which is a full derivation of *the very fat cat eats the mouse* from a two word sentence such as *Tom eats*.

The intuition that language is hierarchically structured, phrases having *immediate constituents* of the type described above, can be extended by the idea of *transformation*.

The idea of transformation is motivated by the observation that sentences which cannot easily be derived from the same simple sentence form in the sense discussed above, are in fact closely related. An example from Chomsky (1957) is the following set of sentences

(7.14) John ate an apple.

(7.15) did John eat an apple?

(7.16) what did John eat?

(7.17) who ate an apple?

All of these may be derived from a single simple base form by applying various *transformations* to the base form, which in this case is

(7.18) John - C - eat + an + apple

Following through these intuitions allowed Chomsky to define his paradigm of transformational generative grammar, which provided a more complete analysis of immediate constituents, and the productive (transformational) relationship between the forms of related sentences. His grammar was also *generative* in that, rather than just seeking to analyse examples, it defined a class of legitimate syntactic structures, based on a context-free grammar which, when forms which did not satisfy certain 'principles' were excluded, was supposed to produce all and only the permissible base forms of natural language.

7.3 Statistical Investigations of the Phrase

One might ask about the relationship between the classes of a *formal* linguistic theory (which are determined by the roles they play in higher order structures), and the classes derived for natural language in the last chapter. Clearly the main theoretical difference is that the previous chapter derived a hierarchical classification of the lexicon, with each word being assigned a particular position in the hierarchy (thus identity of word

determines its classification in this hierarchy). Formal linguistic classifications of words, on the other hand, must allow the class of words to vary dependent on the linguistic context in which the word appears (thus the identity of a word does not determine the class it will be assigned in a linguistic analysis of a sentence in which it appears).

However, when a formal linguistic theory is actually developed, it becomes clear that there is a strong *statistical* relationship between the identity of words and the classes they are assigned in phrasal descriptions of data. It has been found (e.g. Church, 1992) that nearly 90% of word tokens in a text end up being assigned the most likely category of the word of which they are a token.[3] This strong statistical relationship implies that the identity of a word is highly informative (in terms of quantitative information theory) of the class it will eventually be assigned in a structural analysis of the sentence or clause in which it features. It also implies that a model of class assignment where class is a *function* of the identity of the word might be a reasonably good approximation for uncovering further formal linguistic regularities between classes.

The direct analogy of the word level experiments is to split the input corpus into phrasal linguistic units (as opposed to word units), and perform a hierarchical clustering of these in much the same way as described for words. The main problem with this approach is that the phrasal units are not available from raw data — they must be derived.

As a practical issue, the most generally applicable, and easiest, way to investigate phrasal structure is to investigate the distribution of sequences of words which appear in the corpus. One would hope to be able to show that sequences such as 'the big dog' and 'my computer' were more similarly distributed to each other than to sequences such as 'was angry', 'of the people', or 'the name of', since while the first two both can occupy the linguistic role of **noun-phrase**, 'was angry' is a verb phrase, 'of the people' a **prepositional-phrase**, and 'the name of' is not a traditional constituent, but rather it is a noun phrase missing a noun phrase on the right (NP/NP in the terminology of categorial grammar).

The general strategy is as follows: First use the results of the previous chapter to define a partition of the lexicon, assigning each word a unique category. Once a partition has

[3] Using the BROWN tagset of parts of speech, which contains about 30 distinct tags

been defined, the corpus can then be 'tagged' by assigning to each word its respective tag, and sequences of these tags will be the individual phrases being considered. In order to calculate the similarity between two sequences of tags, simply collect statistics about the tags which surround each sequence in the corpus, and in a manner entirely analogous to the word-level analysis presented earlier, cluster on the basis of the similarities between the distribution of these contexts. This approach is, of course, entirely unsupervised.

7.3.1 Partitioning the Lexicon

A procedure to provide a classification of the lexicon was outlined in section 6.3.2. As mentioned then, this partition includes 75% of the words in the original 2000 words which were surveyed, and over 90% of word tokens which were tokens of the original 2000 words occurring in the corpus appear in these 100 categories (since many of the types omitted were relatively infrequent). This list, therefore, defines a functional classification of words into categories. It also defines a mapping between *sequences* of words to sequences of categories, simply by replacing each word in the corpus by its category.

Formally, in terms of chapter 5, a *projection function*, $\mathcal{C}$ (see 5.1.2) is defined so that $\mathcal{C}(w) = \mathcal{C}(w')$ if and only if w and w' are elements of the same class. An additional class was defined so that words are elements of this class if they are elements of no other class (hence this is an 'unknown' class), and the symbols START and PERIOD were assigned classes of their own. The projection Operator, $\mathbf{P}_{\mathcal{C}}$ can then be defined, and this maps a stream of words onto a stream of categories.

We define the *agglutination operator of order n*, $\bowtie_n$, to be a mapping from a stream with alphabet $\mathcal{A}$ to a stream with alphabet $\mathcal{A}^n$ such that

$$\bowtie_n (\Psi)(i) = \langle \Psi(i), \Psi(i+1), ..., \Psi(i+n-1) \rangle$$

Thus this operator can be thought of as producing a stream of fixed length sequences of items from a stream of items. Note that the agglutination operator is formally equivalent to $\prod_{i=0}^{n-1} \mathbf{\Delta}_i(\Psi)$, the product of displaced streams.

Thus $\bowtie_n (\mathbf{P}_{\mathcal{C}}(\Psi))$ is a sequence of n-grams of categories, where Ψ is the original corpus

of words. For instance, if the original corpus is a sequence of numerical digits, say $\Psi = \langle 1, 2, 4, 3, \perp, \perp, \perp, \ldots \rangle$, and the categories given by $\mathcal{C}$ are 'odd' (O) and 'even' (E), then $\bowtie_2 (\mathbf{P}_{\mathcal{C}}(\Psi)) = \langle \langle O, E \rangle, \langle E, E \rangle, \langle E, O \rangle, \langle O, \perp \rangle, \langle \perp, \perp \rangle, \langle \perp, \perp \rangle, \ldots \rangle$.

The strategy followed here is to cluster together sequences of word categories according to the similarities of the contexts in which they appear. We do this rather than attempt to cluster together sequences of words because there are very many fewer sequences of categories than sequences of words. This has two advantages. Firstly, the resources necessary to store the observed context of occurrence of all sequences of words, even of only length 3, would be prohibitive. If there are 2000 words being considered, there are potentially $2000^3 = 8 \times 10^9$ potential trigrams. Even though less than 0.5% of these will actually occur in text, this is still far too large a number of trigrams to conveniently process using the computational resources typically available to researchers. Secondly, and more importantly, the occurrence of most sequences will be very low, thus leading to a large problem if reliable statistics describing their context of appearance are to be collected. Collecting sequences of categories overcomes this problem, to an extent, because each sequence of categories corresponds to a potentially very large number of word level sequences. It is to be hoped that sequences of words identified under this classification usually play similar linguistic roles, so that the form of representation respects linguistic similarity. Whether this is the case can be tested, to some extent, by observing the linguistic tokens of the category sequences, and noting whether the tokens which have identical category-sequence representations often play a similar role in a linguistic analysis of the structures they are part of.

Formally, then, we shall be clustering together sequences on the basis of surrounding context, by analysing contingency tables such as

$$\langle \bowtie_n (\mathbf{P}_{\mathcal{C}}(\Psi)), \boldsymbol{\Delta}_n(\Psi) + \boldsymbol{\Delta}_{n+1}(\Psi) + \boldsymbol{\Delta}_{-2}(\Psi) + \boldsymbol{\Delta}_{-1}(\Psi) \rangle$$

or

$$\langle \bowtie_n (\mathbf{P}_{\mathcal{C}}(\Psi)), \boldsymbol{\Delta}_n(\mathbf{P}_{\mathcal{C}}(\Psi)) + \boldsymbol{\Delta}_{n+1}(\mathbf{P}_{\mathcal{C}}(\Psi)) + \boldsymbol{\Delta}_{-2}(\mathbf{P}_{\mathcal{C}}(\Psi)) + \boldsymbol{\Delta}_{-1}(\mathbf{P}_{\mathcal{C}}(\Psi)) \rangle$$

Note this is a generalisation of the bigram table used in the last chapter to cluster

words on the basis of surrounding context. In that case, $\mathcal{C}$ was the identity function, and $n = 1$. Consequently, this can be thought of as a generalisation of the technique introduced there.

7.4 Experiment

USENET newsgroup articles were preprocessed as described in section 6.1 over a period of about four months, generating a stream of words, Ψ. This was further processed as described above to derive a stream of categories, $\mathbf{C} = \mathbf{P}_{\mathcal{C}}(\Psi)$. Thus $\mathbf{C}$ is a sequence of categories, while Ψ is a sequence of words. The pre-processing symbols START, COMMA, and PERIOD, used to denote punctuation, were assigned a unique category each.

As a first stage, the frequency table

$$[\bowtie_3 (\mathbf{C}) + \bowtie_2 (\mathbf{C}) + \mathbf{C}]$$

was collected, and the 3000 most common items from this table were found, defining the set of items of interest, $\mathbf{I}$. This set, therefore, included the most common sequences of categories of length 1, 2, and 3 from the corpus. The set of items of interest can therefore be split into three disjoint sets — those of length 1, those of length 2, and those of length 3: $\mathbf{I} = \mathbf{I}_1 \cup \mathbf{I}_2 \cup \mathbf{I}_3$. Each of these sets can be used to define a restriction operator, as defined in 5.1.9.

Now, the following contingency tables can be collected and analysed

$$[(\bowtie_3 (\mathbf{C}))_{|\mathbf{I}_3}, \mathbf{\Delta}_3(\mathbf{C}) + \mathbf{\Delta}_4(\mathbf{C}) + \mathbf{\Delta}_{-1}(\mathbf{C}) + \mathbf{\Delta}_{-2}(\mathbf{C})]$$

$$[(\bowtie_2 (\mathbf{C}))_{|\mathbf{I}_2}, \mathbf{\Delta}_2(\mathbf{C}) + \mathbf{\Delta}_3(\mathbf{C}) + \mathbf{\Delta}_{-1}(\mathbf{C}) + \mathbf{\Delta}_{-2}(\mathbf{C})]$$

$$[\mathbf{C}_{|\mathbf{I}_1}, \mathbf{\Delta}_1(\mathbf{C}) + \mathbf{\Delta}_2(\mathbf{C}) + \mathbf{\Delta}_{-1}(\mathbf{C}) + \mathbf{\Delta}_{-2}(\mathbf{C})]$$

These three tables are the contingency tables of the context of occurrence of sequences of length 1, 2, and 3 respectively. If these are arranged into one large contingency table,

the focal items are sequences of lengths 1, 2, and 3, while the peripheral items refer to the values of categories in the context of the sequence. In particular, they refer to the preceding category, last category but one, next category, and next category but one relative to the length of the sequence.

Once collected, the contingency table was normalised, as defined in section 5.1.1, and a hierarchical cluster analysis performed on it using the Spearman Rank Correlation Coefficient as a metric, just as in the natural language data.

7.5 Results

Rather that show small portions of the contingency table, as was done for the word level data, instead the dendrogram was cut at some dissimilarity level, producing a set of sequence classes. In order to show how these sequence classes uncover linguistic structure, first I shall show, for some of the larger classes, their definition in terms of sequences of word level categories. Then I shall show some linguistic 'tokens' of these classes. A sequence of words, $w_1w_2w_3$ is considered a *token* of the sequence $C_iC_jC_k$ just in case the category of w_1 is C_i, the category of w_2 is C_j, and the category of w_3 is C_k. A string of words is a token of a set of sequences (a *sequence class*) just in case it is a token of one of the sequences in the sequence class.

7.5.1 Categories

I shall now present some of the results. First, I shall point out some particularly interesting categories which appear to have been uncovered, from the linguistic point of view, giving in each case both the definition of the category (in terms of sequences of the categories whose definitions can be found in appendix C), with a plausible example for each sequence, and this will be followed by a random sample of some tokens of the sequence category actually found in the text of a draft of this thesis (and consequently not part of the corpus used to derive the categories). After each category, I shall give a tentative linguistic interpretation of the structure of the tokens which has been uncovered by the classification process.

The first category seems to largely correspond to the linguistic entity of the noun phrase.

Category definition

C8 (e.g. *it*); **C8 C3** (e.g. *her status*); **C1 C91 C3** (e.g. *the following section*); **C8 C37 C3** (e.g. *her favourite colour*); **C1 C3 C70** (e.g. *the man himself*); **C1 C3** (e.g. *my idea*); **C1 C3 C3** (e.g. *a street market*); **C1 C23 C3** (e.g. *the federal system*); **C1 C28 C3** (e.g. *the computer network*); **C1 C3 C25** (e.g. *a memory block*); **C1 C23 C25** (e.g. *their commercial approach*); **C1 C25 C3** (e.g. *the state policy*); **C1 C18 C3** (e.g. *a bad society*); **C1 C30 C3** (e.g. *his powerful statement*); **C1 C18 C25** (e.g. *your full name*); **C1 C25** (e.g. *my address*); **C1 C1 C3** (e.g. *such an issue*); **C1 C37 C3** (e.g. *the entire event*); **C1 C37 C25** (e.g. *my favourite place*); **C1 C72 C3** (e.g. *an X event*); **C1 C10 C3** (e.g. *my kill file*); **C1 C28 C25** (e.g. *the math test*); **C1 C17 C3** (e.g. *the published article*); **C1 C48 C3** (e.g. *the first example*); **C1 C41 C3** (e.g. *the other person*); **C1 C48 C25** (e.g. *my last test*); **C1 C16** (e.g. *the messages*); **C1 C18 C16** (e.g. *the legal standards*); **C1 C23 C16** (e.g. *his religious views*); **C1 C3 C16** (e.g. *the network addresses*); **C1 C41 C16** (e.g. *the various questions*); **C1 C37 C16** (e.g. *his own problems*); **C31 C1 C16** (e.g. *all the files*); **C1 C30 C16** (e.g. *the expensive cars*); **C1 C28 C16** (e.g. *some unix machines*); **C1 C50** (e.g. *the answer*); **C1 C23 C50** (e.g. *the actual quote*); **C1 C18 C32** (e.g. *a different one*); **C1 C3 C65** (e.g. *a week today*); **C1 C23 C32** (e.g. *the older one*); **C1 C49** (e.g. *my work*); **C1 C28** (e.g. *the computer*); **C1 C23 C28** (e.g. *their personal insurance*); **C1 C18 C28** (e.g. *a descent individual*); **C1 C79 C3** (e.g. *a middle man*); **C1 C61 C3** (e.g. *a cold room*); **C1 C52 C3** (e.g. *the wrong analysis*); **C1 C73 C3** (e.g. *a stupid argument*); **C1 C27 C3** (e.g. *a better suggestion*); **C1 C18 C10** (e.g. *a significant move*); **C1 C30 C25** (e.g. *an interesting point*); **C1 C18 C59** (e.g. *my full attention*)

Token exemplars

..... and increasing *the apparent size* of the ..... experiments, but *a general* description of ..... which 'stamps' *each article* with an ..... overall within *the newsgroup articles* to include ..... letters, convert *the characters* to lowercase, ... of whether *the article* had already ..... be all *the text* before the ..... not done, *the apparent size* of the ..... began with *the character string* "––", "**", ... important that

this tool works with ..... these tools, *a number* of standard ..... so that *it* takes one ..... differences between *the values* of the ..... fact that *the mother* nodes near ..... learn about *the structure* of the ..... he attributed *each item* a small ..... using GPSG, *a context* free grammar ..... interesting in *this* context that ..... structure of *the real data* and the ..... insomuch as *it* supports inference ..... of presenting *the general* learning paradigm ..... known of *the history* of, and ..... disagree. Defining *a scientific theory* of a ..... what domain *the theory* offers explanations ..... interpreted within *it* (thus consolidating ..... facts about *a domain* for which ..... to expand *the domain* of explanatory ..... use of *the mathematical tools* developed and ..... classification of *the chemical elements* underwent many ..... inspired by *the story* of scientific ..... knowledge, and *this process* might be ..... I define *some terms* for future ..... set, and *all the elements* of this ..... grammars. The *most general* of these ..... immersed in *a society* of users ..... admits only *those examples* seen so

We see from the exemplars that this category does indeed pick out a large number of noun phrases, and partial noun phrases. We note, however, that this simple technique cannot in principle find noun phrases of more than three words in length (e.g. *the general learning paradigm*), and that often it picks out two word sequences which are clearly not noun phrases (e.g. *the general*) (although it might also pick out the three word noun phrases of which they are a part). Also, it often picks out the 'first part' of a long noun phrase, for instance it finds *the story* in *the story of scientific progress.* However, an immediate clause analysis of the phrase *the story of scientific progress* shows that *story of scientific progress* can naturally be thought of as an expansion of its head noun, *story*, in *the story.* Nevertheless, this simple classification uncovers a large number of noun-phrases.

In this regard, it is interesting that object position pronouns are part of this category. Traditional IC analysis analyses noun-phrases as expansions of words like *it*, *she*, and *him.* It is therefore interesting to note that both *it* and *him* are included in this category, although not the accusative personal pronouns such as *he, she, they.* These cluster differently for many reasons (mainly because they precede verbs more than ordinary nouns and aren't used much in prepositional phrases).

It is interesting to note that words of category C10, typically verbs, operate as nouns within this category, occupying the same positions as typical nouns (e.g. C3). This also applies to the ambiguous category, C25, which contains words which are often used as nouns, and often used as verbs.

The second category corresponds to various forms of the verb *to be*. Linguistically, this is a special verb because it can be used attributively (*Sade* **is** *a woman*), or as a helper verb (e.g. *John* **will be** *going to London*), in a variety of tenses, and taking agreement with its subject.

Category Definition

C4 (e.g. *is*); **C4 C22** (e.g. *was simply*); **C4 C20** (e.g. *are not*); **C4 C22 C20** (e.g. *am probably not*); **C9 C19** (e.g. *will be*); **C9 C20 C19** (e.g. *might not be*); **C9 C22 C19** (e.g. *could just be*); **C9 C14 C19** (e.g. *might have been*); **C53 C7 C19** (e.g. *seems to be*); **C24 C22 C19** (e.g. *had never been*); **C4 C20 C22** (e.g. *are not really*); **C4 C2 C57** (e.g. *are in fact*); **C24 C7 C19** (e.g. *has to be*); **C4 C7 C19** (e.g. *am to be*); **C22 C9 C19** (e.g. *just shouldn't be*); **C24 C19** (e.g. *has been*);

Token Exemplars

words long *is* an existence .....evidence. There *are* several possible .....space which *is* either finite, ...If this *is* the case, ...when this *is* done, generalisation .....of language *will be* complete, and .....the grammar *will have been* learned. This .....possible grammars *has been* vastly reduced .....the grammar *has been* made much .....because one *is* no longer .....so enumerations *are* finite. It .....finite. It *may still be* that negative .....negative evidence *is* required (for .....language which *is* a subset .....alternative method *is* to use .....that language *is* rarely used .....what it *is* referring to, ...between what *is* spoken, and .....referred to,*can be* used to .....them. This *is* discussed more .....the problem *might be* to make

As can be seen, the tokens are all of parts of the verb *to be*, possibly including adverbial modifiers. Moreover, the coverage is fairly complete — there are few parts of the verb *to be* which are not covered by the category. One glaring exception, however, is the bare form of the verb itself. The sequence **to be** is not included in this category. In

fact, it forms a category on its own, being significantly differently distributed from this category. In fact, it clusters rather more similarly to sequences like **to go**, and other infinitive verb parts. Nevertheless, there is a very high correlation between the tokens of language associated with this category, and the linguistic notion of parts of the verb *to be*.

The next category appears to be a prototypical prepositional phrase.

Category Definition

C2 C16 e.g. *of games*; **C2 C3 C16** e.g. *for research ideas*; **C2 C28 C16** e.g. *of pc utilities*; **C2 C41 C16** e.g. *in other words*; **C2 C18 C16** e.g. *with strong views*; **C2 C23 C16** e.g. *on older machines*; **C2 C31 C16** e.g. *about both articles*; **C2 C1 C3** e.g. *into the battle*; **C2 C1 C25** e.g. *with my address*; **C2 C8 C3** e.g. *against this background*; **C2 C11 C3** e.g. *towards that conclusion*; **C2 C1 C85** e.g. *in the USA*; **C2 C1 C49** e.g. *of my work*; **C2 C32 C3** e.g. *in one way*; **C2 C1 C50** e.g. *in the show*; **C2 C8 C25** e.g. *with this filter*; **C2 C1 C16** e.g. *among my messages*; **C2 C1 C29** e.g. *of the people*; **C2 C1 C77** e.g. *with my hands*; **C2 C1 C42** e.g. *at his home*; **C2 C1 C8** e.g. *in the US*; **C7 C1 C3** e.g. *to the subject*; **C7 C1 C25** e.g. *to the point*; **C7 C1 C16** e.g. *to my experiences*; **C7 C8 C3** e.g. *to this directory*; **C7 C1 C29** e.g. *to their friends*; **C2 C3** e.g. *of oil*; **C2 C3 C3** e.g. *in light traffic*; **C2 C18 C3** e.g. *with similar performance*; **C2 C23 C3** e.g. *in second place*; **C2 C28 C3** e.g. *for general information*; **C2 C28** e.g. *without insurance*; **C2 C1 C28** e.g. *from the internet*; **C2 C1 C92** e.g. *with my apple*; **C2 C1 C66** e.g. *in no science*;

Token Exemplars

(this varies *from case* to case ..... The answer *to the questions* case folding ..... the set *of all words* in the ..... machine acquisition *of language* from pure ..... the acquisition *of language* from real ..... a subset *of language* generated by ..... learner's perception *of the situation* being referred ..... a corpus *of positive examples* of a ..... that information *about the situation* being described ..... a view *of natural language* processing compatable ..... the intuitions *of language* users, the ..... underlying structure *of the rules* and representations ..... lexical hierarchy *of words* which, it

..... phrase up *in a table* and the ..... acquire some *of the rules* of grammar ..... the sum *of the description* of the ..... two strings *of words* are identical ..... this being *with reference* to other ..... label associated *with the word* in question, ... statistical models *of their training* data. In ..... the structure *of the source* is, what ..... optimal predictions *of the output* vector under ..... first model *of data* we shall ..... vector structure *of the data* by essentially ..... the piece *of paper* having seen ..... the inference *from model* to the ..... the nature *of the language* model. Marcus ..... *E* occurs *to the number* of times ..... the probability *to a sequence* of characters. ..... and weaknesses *of this model* and by ..... higher elements *of structure* in natural ..... in terms *of classes* of the ..... tape recording *of a speech* act, but ..... includes 75 *of the words* in the ..... one relative *to the length* of the ..... the variation *of light* intensity originally ..... for descriptions *of the world* outside the ..... the groove; *in the case* of the ..... be added *to this analysis* of computation, ... of view *of a system* which learns, ... between representations *on the basis* of some ..... the investigation *of this point* where language ..... linguistic domains, *on the basis* of many ..... Z *in the case* of a ..... discrete sequence *of items* such as ..... the values *of the product* stream occur, ... contingency tables *in terms* of old ..... the number *of items* fitting either ..... the distances *at every stage* in the ..... all pairs *of points* is 1 .....

As can be seen, nearly all the tokens picked out are prepositional phrases, or the start of prepositional phrases including the head noun of the associated noun-phrase.

7.5.2 Discussion

A considerable amount of linguistically perspicuous structure within small sequences of words can be elucidated by the unsupervised techniques used in this thesis. However, although the results are encouraging, there are some signs that this is close to the limit of the efficacy of the entirely unsupervised approach. Whereas in the case of deriving word classes in the previous chapter nearly all the categories derived were linguistically perspicuous, here there are several categories which have little apparent linguistic motivation, as well as several categories which are clearly mixtures of more than one linguistic type.

However, these categories are not useless, because they can be used to uncover structure at a higher level, since a mixed category, even though not linguistically coherent on its own, may become so given additional context. For instance, in the word-level categories there is a category of words which are sometimes used as nouns, and sometimes used as verbs. However, if preceded by a determiner, they are almost always used as nouns, and if preceded by an auxiliary, almost always as verbs. Therefore they become part of a linguistically coherent category when given the right context.

7.6 Theory of Higher Level Sequences

This section discusses some results from experiments in finding structure in language over longer distances than just sequences of up to 3 words using the unsupervised, non-parametric techniques described in this thesis. This is a very much harder problem than just considering the short sequences as described above. In fact, the short sequences described above seem to be at the limit of the efficacy of this approach to uncovering significant syntactic structure, greater structure seeming to need more linguistic knowledge to be effective at uncovering structure.

Once sequences of length 1, 2, and 3 have been classified as above, it is possible to exploit this classification to learn structure in longer sequences. This proceeds roughly as follows: First classify the sequences into categories according to the similarity of their distribution, just as was done for words. Next, it is possible to consider sequences of *these* categories, and classify them according to the similarity of distribution of the contexts in which they occur.

Effectively, what we have now is a simple parsing scheme of the following form. First, we have the basic, *C-level*, categories of the form

C1 → *the*

C1 → *my*

C3 → *house*

C3 → *person*

C2 → *of*

C2 → *in*

C8 → *me*

C8 → *him*

⋮ ⋮ ⋮

C? → Any word not in any other category.

The set of C-level categories is the alphabet of the stream **C** above, and shall be denoted C.

and then we have short sequence, *X-level*, categories of the form

X30 → C1 C3

X30 → C1 C3 C3

X30 → C8

X36 → C3

X36 → C3 C3

⋮ ⋮ ⋮

X? → Any short sequence not in any other category.

The set of *X-level* categories will be an alphabet used below, where it shall be denoted X.

By defining a very simple context-free grammar of the form

S → X1 | X2 | X3 | ... | X?

S → S S

it is now possible to parse *any* sequence of words into chunks of length 1, 2, and 3. Each sequence of 1, 2, or 3 words will be called a *short sequence*. It is also clear that a sequence might be split into short sequences in very many ways. For example, the sequence

The big black dog

can be structured into labelled bracketings of short sequences in 7 ways: (X81: The)(X32: big)(X32: black)(X36: dog); (X81: The big)(X32: black)(X36: dog); (X81: the)(X32: big black)(X36: dog); (X81: the)(X32: big)(X36: black dog); (X81: The big black)(X36: dog); (X81: the)(X36: big black dog); (X81: The big)(X36: black dog). Each labelled bracketing, or *parse*, corresponds to a sequence of X categories: in this case, the sequences are X81 X32 X32 X36; X81 X32 X36; X81 X32 X36; X81 X32 X36; X81 X36; X82 X36; X81 X36.

Since the initial segment of any sequence must be of length 1, 2, or 3, it is clear that the number of parses[4] must be f_n, where f_n is defined recursively as follows:

f_1	$=$	1	(a)
f_2	$=$	2	$(a)(b), (ab)$
f_3	$=$	4	$(a)(b)(c), (ab)(c), (a)(bc), (abc)$
f_{n+3}	$=$	$f_{n+2} + f_{n+1} + f_n$	$(a)[bc\mathbf{s}], (ab)[c\mathbf{s}], (abc)[\mathbf{s}]$

The final column justifies the formula of the row for the case of a sequence of the form $abc\mathbf{s}$, where $\mathbf{s}$ is a sequence of length n. Round brackets around a sequence denote one particular parse, while square brackets denote the set of all possible parses.

The first three rows are the number of possible parses of sequences of length 1, 2, and 3 respectively, and the final row expresses the fact that the number of possible parses of a sequence of length greater than 3 is the sum over the number of ways the initial segment can be parsed of the number of ways the rest can be parsed, which is defined to be f_n.

Thus, from a sequence of words it is possible to find a set of parses, and from this set, it is possible to find a set of sequences of **X**-level categories. This set of sequences of **X**-level categories will be the basis for a higher order classification. The situation is displayed in figure 7.2, for one particular possible parse of the sentence "How can an

[4]Distinct labelled bracketings of the sequence into short sequences are counted as distinct parses. Since every **C**-level sequence falls into precisely one of the **X**-level categories (including the unknown sequence, X?), the number of parses is no greater than the number of bracketings, since a bracketing of the sequence determines a labelled bracketing, or parse. It is clear that it can be no less than the number of bracketings, since distinct bracketings necessarily give rise to distinct labelled bracketings.

agent come to acquire a theory of ...". In this figure, different groupings of sequences at the **C** level give rise to different sequences of **X**-level categories, corresponding to different parses of the **C** level stream. This figure shows that the statistical analysis which derived the **C**-level categories and the **X**-level categories is entirely analogous to that used to derive the **S**-level categories, the only difference being that the parsing of the **X**-level stream is complicated by the non-determinism in the mapping between the **C**-level stream and the **X**-level stream.

7.6.1 Parsing Streams

This section provides a formal definition of streams used to perform the statistical analysis on sequences of **X**-level categories. From figure 7.2, it is possible to see that every **X**-level category corresponds to a **C**-level sequence, and a sequence of words, by projection. For what follows, a **C**-level sequence means a sequence of **C**-level categories of length 1, 2, or 3.

Consider a portion of a long stream of words, Ψ, first categorised as described above to form a stream of *categories*, $\mathbf{C} = \ldots c_{n-6}c_{n-5}\ldots c_n c_{n+1}\ldots c_{n+8}c_{n+9}\ldots$. It is possible to find the set of all possible parses of this string into **X**-level categories which involve a **C**-level sequence starting at c_n. Recall that each parse gives rise to a sequence of **X**-level categories, and we shall be interested in directly applying the techniques described earlier to sequences of length 1 and 2 of *these* sequences. Such sequences of **X**-level categories will be called **X**-level focal sequences, and shall later be the projection of **S**-level categories.

For example, one possible parse of the sequence might have a **C**-level sequence starting at c_n and finishing at c_{n+2}, followed by a **C**-level sequence starting at c_{n+3}, and finishing at c_{n+4} (a sequence of length 3 followed by a sequence of length 2). This gives rise to a sequence of two **X**-level categories. To apply the techniques described above, it will be necessary to find statistics about the **X**-level categories surrounding this **X**-level focal sequence.

Thus, for each **X**-level focal sequence, we must be able to determine the set of **C**-level

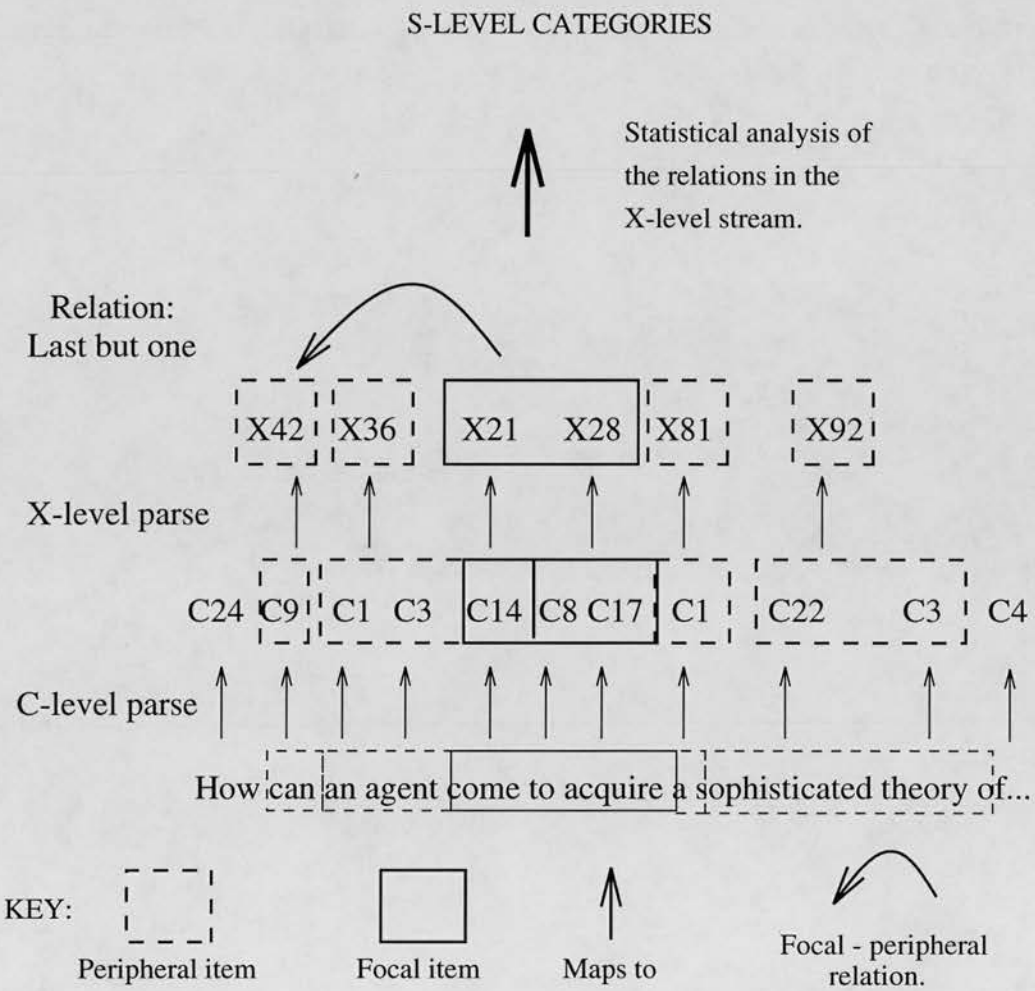


Figure 7.2: This figure shows one particular **X**-level analysis of a sentence surrounding the words "come to acquire". At the top level, the focal items are **X**-level categories, and sequences of two **X**-level categories, and the peripheral relations are identical to the word level analysis, and the **C**-level analysis. Once the data from these relationships has been analysed, the clustering of sequences of **X**-level categories gives rise to **S**-level categories in the same way that the clustering of sequences of **C**-level categories yielded **X**-level categories.

sequences (and hence the **X**-level *categories*) around it — the two previous **C**-level sequences, and two following **C**-level sequences. For instance, suppose the **X**-level focal sequence we are considering starts at c_n, and finishes at c_{n+4}. Then it is possible to find all possible immediately following **X**-level categories (i.e. those **C**-level sequences which start on c_{n+5}), and all possible 'next but one' categories (i.e. those sequences starting between c_{n+6} and c_{n+8}). Similarly, it is possible to find all possible immediately preceding **X**-level categories, and all 'last but one' **X**-level categories. In fact, there are three possible 'immediately following' categories — one for each sequence of **C**-level categories of length 1, 2, and 3, and nine possible 'next but one' categories — one of each length starting at one of three possible starting points. Similarly for the previous and 'last but one' **X**-level categories. For any long sequence, there are $3^4 = 81$ possible parses of the four surrounding **C**-level categories, since each surrounding **C**-level category may independently be of length 1, 2, or 3, and a different choice for the length of any one yields distinct sets of possible parses.

Thus, we can define a set of functions between sequences of words and focal-peripheral pairs of **X**-level categories which picks out the relations 'Focal sequence - Next **C**-level sequence', 'Focal sequence - Next **C**-level sequence but one', 'Focal sequence - Last **C**-level sequence', 'Focal sequence - Last **C**-level sequence but one'. Let us consider one such relation — 'Focal sequence - Next **C**-level sequence but one'.

A Focal sequence is defined to be either a **C**-level sequence, or an agglutination of two **C**-level sequences, just as the sequences considered earlier were either a category, an agglutination of two categories, or an agglutination of three categories.

Before proceeding, I shall define some useful notation which allows us to refer to the structure of **C**-level sequences parsed into a sequence **X**-level categories. The parsed structures we shall be interested in are all of the form

(7.19) $\ldots x_{-2}x_{-1}sx_1x_2\ldots$

which are derived from category sequences of the form

(7.20) $\ldots c_{n-6}c_{n-5}\ldots c_nc_{n+1}\ldots c_{n+10}c_{n+11}\ldots$

where the first category of the sequence s is c_n, and where x_i is the ith **X**-level category following the focal sequence, s, which projects to a sequence of **C**-level categories of length 1, 2, or 3.

Each focal sequence has multi-sets of 3 possible immediately preceding and following **X**-level categories, since the immediately following element, for instance, must start immediately after the focal sequence, but can have a length of 1, 2, or 3 **C**-level categories, and the same is true for the immediately preceding **X**-level category. It has multi-sets of 9 'next but one' and 'last but one' **X**-level categories, since there are three possible starting places for these categories, and for each starting point the **X**-level category can extend to 1, 2, or 3 **C**-level categories.

We write $X_{c_1c_2c_3}$ for the single **X**-level category which expends to $c_1c_2c_3$. Similarly for $X_{c_1c_2}$, and X_c. Thus the multi-set of **X**-level categories immediately following a focal sequence s which, when projected to the **C**-level, ends at c_m, consists of three elements — $\{X_{c_{m+1}}, X_{c_{m+1}c_{m+2}}, X_{c_{m+1}c_{m+2}c_{m+3}}\}$. We define five operators — the first parses a sequence of up to 6 **C**-level categories into a multi-set of one and two long sequences **X**-level categories (which correspond to the values of the focal sequence), and the remaining four return the multi-sets of surrounding categories for a given focal sequence, one for each of the four relations "next sequence", "previous sequence", "next sequence but one", and "previous sequence but one".

We define the *focal sequence operator*, F, to be an operator which takes an index for the corpus (corresponding to the start of the sequence), and an integer (corresponding to the length of the focal sequence), and returns the multi-set of **X**-level parses of this sequence. If we superscript the length of the focal sequence (to avoid confusion), then we can write

$$F^4(\Psi)(n) = \{\langle X_{c_{n+4}}, X_{c_{n+5}c_{n+6}c_{n+7}}\rangle, \langle X_{c_{n+4}c_{n+5}}, X_{c_{n+6}c_{n+7}}\rangle, \langle X_{c_{n+4}c_{n+5}c_{n+6}}, X_{c_{n+7}}\rangle\}$$

The corpus, Ψ, can either be considered as a word stream, or as a stream of **C**-level categories, since the map between the two is deterministic. Similarly we can define F^1, F^2, F^3, F^5, and F^6, allowing any sequence from length 1 to 6 to be a focal sequence, at any point in the corpus. Similarly, we define the "previous category" operator to

be a function which takes a corpus, and an index which marks the start of the focal category (i), and the length of the focal sequence (l), and returns the multiset of previous categories. If we call this operator D^l_{-1}, for $l \in \{1, \ldots, 6\}$, then

$$D^l_{-1}(\Psi)(i) = \{X_{c_{i-1}}, X_{c_{i-2}c_{i-1}}, X_{c_{i-3}c_{i-2}c_{i-1}}\}$$

Similarly, we can define the "following category" operators (D^l_1), the "next but one category" operators (D^l_2), and the "last but one category" operators (D^l_{-2}).

Now we are in a position to derive the **X**-level streams.

Definition 7.6.1 (X-level multi-stream) *Let Ψ be a corpus with the integers, $\mathbf{Z}$, as an index set. We derive the following five streams.*

$\mathbf{F}^M$, *the stream of focal items, is a multi-stream defined by*

$$\mathbf{F}^M(6i + l) = F^l(\Psi)(i)$$

$\mathbf{D}^M_i$, *the multi-streams of immediately following category ($i = 1$), immediately preceding category ($i = -1$), next category but one ($i = 2$), and last category but one ($i = -2$), are defined by*

$$\mathbf{D}^M_i(6n + l) = D^l_i(\Psi)(n)$$

Thus for any finite portion of Ψ with indices in the range $0, 1, \ldots, n-1$, these four sequences will have indices in the range $1, 2, \ldots, 6n$. So each element of the corpus gives 6 elements of the **X**-level multi-streams. The contingency table which is collected is simply

$$[\mathbf{F}^M, \mathbf{D}^M_{-2} + \mathbf{D}^M_{-1} + \mathbf{D}^M_1 + \mathbf{D}^M_2]$$

as defined in definition 5.1.8, which is entirely analogous to the word level case.

7.7 Experimental Details

C-level sequences of length 1, 2, and 3 were classified as above into **X**-level categories, and sequences of length 1 and 2 of these **X**-level sequences were collected, and the 3000 most common such sequences formed the set of focal items for this experiment. The **X**-level categories were selected as the peripheral items. The corpus was then parsed in the manner described above, a dendrogram created, and this dendrogram was used to form longer sequence categories in just the same way that the dendrogram of words was cut to form word classes, and the dendrogram of **C**-level sequences was cut to form classes of **C**-level sequences. The resulting classification is called the **S**-level categorisation.

The results will be presented in the same form as the results for the **C**-level sequences were presented, concentrating on proto-sentences, proto-noun-phrases, and proto-prepositional phrases, except that in each case no category definition will be given. This is because without a full definition of all the **X**-level categories, **S**-level definitions would be meaningless.

Proto-Sentences

A proto-sentence is a phrase which could reasonably be thought to be a candidate sentence if parsed out of context. For instance, the phrase

(7.21) The man ate

would be such a sentence, even if it occurred in the context of

(7.22) The man ate the apple.

in which it would not be assigned the role of *sentence*. Also, because of natural language effects like ellipsis, NP movement, and the like, many sequences may be analysed as sentences which don't themselves stand alone as candidate sentences.

Here are some random examples of proto-sentences from the category S14, whose tokens seem to be proto-sentences. The examples were selected at random from newsgroup

articles not part of the corpus used to generate the various components of the system. The random example selection process is biased in favour of displaying longer sequences.

... START just *what is a context* free grammar PERIOD ... START but *it might be a good idea* ak to give ... START *that's a different story* PERIOD ... and *you see a problem* with this let me know ... START *you will also receive a copy* of robert's own morphology analysis ... someone pointed out to me *that there was an error* in testdcg PERIOD ... START *the world isn't perfect* in this case COMMA ... START if *you start out* with advanced targets ... or *it really was lost* in translation PERIOD ... START although *it does have a german title* PERIOD ... START *we are looking* for new accommodation ... born well off and annoyed *the government won't let them* take care ... START we thought *it would be a good idea* to get ... START also *i did notice it* was the final ... START does anyone know where *i can get the book* COMMA ... START *you can actually see it* COMMA ... START if *you need more information* about them COMMA ... START i know *this information is available* via the tel program COMMA ... if *they were picked up* as soon as possible PERIOD ... START perhaps *we could hold some events* together PERIOD ... if *you carry them* in a bag or pocket ... START end launch *i just received my copy* of masque and COMMA ... just in case *you were found out* PERIOD ... START *i just don't want it* to continue at that time ... becuase *i am used* to beaver stadium COMMA ... START *i'm a full time student* and don't have a whole ... when *you pull his head* up COMMA ... START *i found out it* was corrupted PERIOD ... START if *you can't tell the difference* in style between ... START if *you kill another player* COMMA ... START but *i don't remember the name* PERIOD ... START *it had a small hole* and some white COMMA ... less to worry about when *you're small* PERIOD ... START *he told me* his COMMA ... START if *you put a city* down in panama COMMA ... the benefits of *it's products* COMMA ... START *i agree* entirely PERIOD ... START *you should check them out* PERIOD ... START *you can get the source code* for COMMA ... the fact that *i used this* newsgroup as a forum ... START *i forget the name* COMMA ... START *some things are added* very late ... car has been running a *while there is no problem* PERIOD ... START *my piece is small* but effective COMMA ... START *this is the place* for it PERIOD ... START so maybe *this wouldn't be a problem here* PERIOD ... START just because *every country does it*

does not make

It is quite apparent that the tokens accord quite closely with the linguistic entity of the sentence, although often the procedure identifies only the first portion of the sentence. Often, if the verb takes two compliments, only the first is included, and for sentential compliments, we find the problem that the first NP of the sentence is often taken as the compliment of the verb (e.g. START *i found out it* was corrupted PERIOD). Also, since context is not taken into account, we have the spurious *car has been running a* while there is no problem *PERIOD.* Note, that the statistical regularity which finds sentences is far more subtle than just looking for sequences which typically occur between the special tags START and PERIOD, as a glance at the data above will show — most of the proto-sentences identified do not occur between these symbols (although clearly, these symbols *are* indicative of sentence start and termination.

Prepositional Phrase

Here is a random selection of token from a category whose tokens largely correspond to prepositional phrases.

... will all settle *out* in the end ... START to get *out of this state* i believe ... i will collate them *into a form* where they ... more information *about formal language theory* as it applies ... answers *to those questions* COMMA ... the study *of language and information* PERIOD ... file them *in the appropriate box* PERIOD ... parameter argument *to a function* which changes its behavior ... START *in school french* the infinitive was the name ... START take *it out* PERIOD ... START *by the way* COMMA ... most *of the terms* used in theoretical linguistics ... START depending *on its argument structure* COMMA ... the value *of a variable* and you don't ... you see a problem *with this* let me know ... quantitative measures *of program performance* a COMMA ... START *on the basis* of this exercise COMMA ... the beginning *of the file* PERIOD ... START *in other words* COMMA ... topics relevant *to the development* of modern ... inviting all netters *in general* and egyptian netters ... the creation *of such a news group* was discussed ... START *for this* reason COMMA ... to discuss dr v *on usenet* COMMA ... START *in areas of political rights* COMMA ... START especiallly one run *on the basis of religious law* PERIOD ... any evidence to back it *up*

PERIOD ... such apparent exceptions *to such a rule* PERIOD ... the few first words on *his way back* home PERIOD ... if there is a rule *regarding this issue* COMMA ... we can come *to a simple conclusion* PERIOD ...

The tokens here are almost exclusively either full prepositional phrases, or the first part of the prepositional phrase including the head noun of the rest of the prepositional phrase.

Noun Phrases

This category includes the **C**-level noun phrases described above, and more complicated constructions such as **Det NBAR PP**, as in *the child of a woman* or *a piece of paper*, as well as longer sequences of the type **D (A*) N***, but not sequences such as *the man who I saw yesterday*, possibly because these are typically too long to be considered.

... START *the reason* for this is that ... an expert in *such questions* PERIOD ... the currency of *a moral law* PERIOD ... START *the problem with it* is ... START anyone with *a more accurate memory* COMMA ... element of *the real number system* this equation proves ... START *the article* i read ... tell you that *it* is wrong PERIOD ... START in *many cases the option* to do so ... not wish to get into *a discussion on this* here COMMA ... whether *the child of a woman* converted by a ... the one i used in *the us* COMMA ... this wouldn't be *a problem here* PERIOD ... an enemy soldier placing *a gun* to one's head PERIOD ... START i supported *the six day war* COMMA ... extrapolating from *his behavior during his life* it is ... START as to *his ideas about the rights* of non jews ... we already substituted *the four letter name* with adon ... to use *it for no reason* COMMA ... we can't destroy *a piece of paper* containing either name ... remember to ask *someone at the post* PERIOD ... could provide *some sources for your last statement* PERIOD

Tensed Verb Phrases

This category includes a predomination of verb-phrases. The verb can appear in many forms, including being embedded in a complicated phrase (e.g. *would like to hear about it*). It includes verbal particles (e.g. *could buy off them*), and adverbial forms (e.g. *is also available*). The complement can either be not present (as in intransitive verbs), or

can be a NP, a PP, or an adjectival complement.

... START we *would like to hear about it* PERIOD ... START the letter *came* from belgium PERIOD ... how much damage *has been done to the engine* COMMA ... START we thought it *would be a good idea* to get in touch ... an adaptor that i *could buy off them* for some reasonable ... START but i *wouldn't worry about them* PERIOD ... START at least you *can eat it* PERIOD ... a friend who *is doing* a tefl course ... START this talk *will describe the signal project* COMMA ... the common terminal rooms *are used* pretty much all day ... the heater *will continue* to release heat ... START a beginning *is being made* by a special commission ... START the projects *were carried out by* both governmental and nongovernmental organizations ... START and equipment *needs of the data* management unit ... the design process *will begin this month* PERIOD ... if this *is not the case* PERIOD ... only ai soc members *can come to this* one PERIOD ... START when errors *are discovered in a knowledge base* COMMA ... it *was possible* to do COMMA ... it *will be necessary* to do COMMA ... START a concrete example *will be given* concerning polish ... maybe we *could write a paper* on this PERIOD ... suppose we *are given* some measurements ... START i don't think spark *is the problem* COMMA ... START a car *is also available* PERIOD ... he also *has a minor role* as a crazed sas leader ... but you *don't have to try* to make it dangerous ... START i *can go over* to alt PERIOD ... START this *is a local news story* that really ... START before i *went* to this isle ... assert that snow *is white* if snow is not white ... i just heard that superman *is going to die* PERIOD ... skin cancer rate increase *will occur* COMMA ... START polystyrene recycling *does not save energy* PERIOD ... scale over which this occurs *is the length of time* it takes to ... START sea ice *is much* fresher than sea water COMMA ... START it *turns* out COMMA ... START but it *runs into the problem* that precipitation ... these plutonium shipments *did come* a cropper in a storm ... that it *was a silly idea* PERIOD ... the spacecraft *must be made of any material* COMMA ... START this system *has to be big* COMMA

Verb Phrase II

This category includes past tense verb phrases of the form **V NP**, **V Prep**, **V PP**, together with sub-classes of adverbs around the verb, and verbal particles.

... where rainbow warrior *received the letter* from mr PERIOD ... unload might be *carried out* PERIOD ... with shikishima which *kept the direction and speed* PERIOD ... or popular science *had an issue* or two about that ... the gm special labs guys *made it* PERIOD ... START i *thought it* might be of interest ... they had *used it* PERIOD ... START open loop steam *had another problem* COMMA ... how greenpeace could have *done anything* about it PERIOD ... START greenpeace *has many ways* in which they could improve ... do not seem to have *taken this on board* PERIOD ... regarding the post i've *sent out* earlier on COMMA ... so strongly *supported the argument* that when we ... methods can be *called into question* PERIOD ... so i have *checked into it* PERIOD ... with hydrogen *added to make them* less stable COMMA ... START *used for products* that have replaced cfcs with ... after defects were *found in the fuel* channel control valves PERIOD ... START the experts *found out* that in bohunice COMMA ... whoever *posted this* is strongly agin the ... the pressure *used by air* liquification plants COMMA ... they *found out* it was a kgb spy ... but it is *also made up* of bee faeces PERIOD ... that need to be *used by several programs* COMMA ... who hadn't *written their own versions* of all of these already ... START only mathematica *had the resources* we needed PERIOD ... START *gives an error* message and abort's PERIOD

Non-Finite Verb Phrases

... START million and one dollars *to use the resource* COMMA ... concerned not *to do it* COMMA ... START *to start* all over again ... should be used *to bring the government* of france to justice ... and adjusting the tax *to reach it* PERIOD ... the right way *to go* PERIOD ... will offer opportunities *to learn of the special character* and function of southern appalachian ... START *to join us* iale COMM ... to know how *to fly* PERIOD ... they hope *to find better* opportunities ... which are already hard *to come* by COMMA ... in order *to use them* in their articles PERIOD ... should be able *to find a list* of distributors ... but enough *to get the idea* of what is possible ... my cat likes *to play with them* PERIOD ... START how *to go about it* PERIOD ... which i would like *to use for frame by frame* recording ... is it illegal *to sell something* that doesn't have ... if someone wants *to market a pc* COMMA

N-Bar phrases

There are determinerless noun phrases. It should be noted that it includes coordinations of such phrases (as it should) (e.g. *science and technology policy*, *views and opinions*, and so on). The structures which seem to be covered by this class include:

N; A N; A A N; A N N; N PP; (A CONJ A) N; A (N CONJ N); N N N; N CONJ N; and so on.

... suspended in *air* for one minute PERIOD ... damaging uses are of *small economic value* COMMA ... i'd like to *point* out that ... impact on short term *economic value* PERIOD ... START nasa goddard *space flight center* hello everyone ... to challenge the *view* that reprocessing spent fuel ... manufacture of *electrical and electronic equipment* COMMA ... START office of *science and technology policy* COMMA ... on microwave *theory and technology* COMMA ... the *production of low level* em fields ... damaging thermal *effects to the human body* PERIOD ... to have the *necessary economic power* to fully ... to follow the *operation of this* home to see if ... START a *free market in air* would ... some things that you need *common sense* in PERIOD ... i watched a children's *educational tv program* PERIOD ... START the *ice* is basically pure water PERIOD ... START the ice is basically *pure water* PERIOD ... every couple would have one *child the population* is reduced ... to be a *reference* book by COMMA ... to be a *reference book* by COMMA ... to find this *book* PERIOD ... would not infringe privately owned *rights* PERIOD ... START no *specific reference* constitutes ... START no specific *reference* constitutes or implies ... by the united *states* government ... by the united *states government* or the idaho ... by the united states *government* or the idaho ... or the idaho *national engineering* laboratory ... the idaho national *engineering* laboratory PERIOD ... START the *views* and opinions expressed ... START the *views and opinions* expressed ... START the views and *opinions* expressed herein ... START although with prompt *medical attention* COMMA ... START they have established *standards for various types* of domestic and foreign cars ... can get a free macintosh *program* to use ... START use the *user name* COMMA

Incoherent I

Finally, I shall give some examples of some relatively linguistically incoherent categories[5].

[5]Again, although this category and the next do not conform to standard linguistic constituency, it

The first one contains a mixture of categories which might be of the categorial category NP, NP/NP, or small clauses. For instance, *his way to court* might be part of *his way to court a woman.* However, the category is relatively incoherent at this level (although it might be that it becomes a unit of a more coherent category at a still higher level). They often include words typically used as nouns being used as verbs, and vice-versa. This might give rise to unreliable statistics for this category, since the sample size of words used in this way will often be small.

... he was just on *his way to court* for such a situation PERIOD ... this is *a call* for votes COMMA ... i don't see *any mention of* any class ... START you can watch *this happen* COMMA ... does a device which performs *these functions exist* PERIOD ... START i am building *a device to monitor* multiple phone lines for ... the crystals may moose *the ability to display* information ... START is there *any easy way to make* this happen PERIOD ... i'd rather not take *the time to make* a high power hack ... want the true color of *the subject getting* to the film PERIOD ... to hook mysoundblaster to *the phone line without getting* in trouble with the phone ... how can one detect *the call* charge pulses on a telephone ... START *a small price to pay* for good tunes in the ... and telling *me to leave* PERIOD ... START is there *any way to go* the other way PERIOD ... there is *no way to tell* which channel a part of ... is there *some way to create* COMMA ... START they manufacture *an item called* the micro dac ... START and now *the catch* has finally come PERIOD ... i will try to get *the text file to do* it if anyone is interested ... START *a simple high pass* filter COMMA ... START could *the original poster send me* more info ... printers generate something that cause *me to feel* very bad ... START does *this look* correct PERIOD ... or phone number for *a place to get* small COMMA ... START there are *several ways of doing* this COMMA ... and use *the output to drive* the mosfets PERIOD ... you would feel *the pull* on your car keys ... START *some get* caught COMMA ... START *some get by* PERIOD ... opposition to *the change* in current PERIOD ... opposition to *the change in* current PERIOD ... and a scope is *a great way to see* these effects in action

---

should not be assumed that standard linguistic constituency is what the statistics should show. Many of the examples in this category are of the form "Subject Verb", which is a constituent under a flexible constituency account.

Incoherent II

The second incoherent category includes subject position pronouns, and so it might seem rather strange that it is indeed linguistically incoherent, especially when one considers that the object position pronouns cluster with other noun phrases. However, although it would be hard to give this category a standard linguistic interpretation, many of the examples presented here would be assigned the category S/(S\ NP) or just simply NP.

... START if *this isn't what you* mean COMMA ... what *you* mean COMMA ... START *it doesn't* have ... START *can anyone* suggest a way ... START *i have not* had so much luck ... START when *i* tried to order items ... START when *i tried to* order items ... START *this way you will* have many ... START this way *you* will have many catalogs ... START this way *you will* have many catalogs ... START so *the goal is to* present the pc with ... START *does anyone* have a working device ... START *i would like to* have a videocamera COMMA ... *i would* like to have a videocamera ... START or starts in *the right direction will* will be greatly appreciated PERIOD ... START *where should i* look for more information

7.7.1 Transformations

Recall that one of the motivations behind Chomsky's original proposal to analyse sentences using transformational grammar was that such a formalism was necessary to find the relations between different forms of sentences (e.g. *John ate an apple.*, *What did John eat?*, *Who ate an apple?*, and *Did John eat an apple?*). He argued that it was hard to see how immediate constituent analysis, and its derivatives, could explain the productive nature of such sentence forms. It is therefore an interesting question to ask whether such regularities can be captured by the system described above.

Since *apple* appears in USENET articles mainly as the name of a type of computer, and as the name of a computer company, Chomsky's original example will not do, and has to be modified. Also proper names are relatively rare, so the 'most frequent first' techniques used here won't include many forms with proper names. Consequently, the example was changed, and the following sentences were examined:

(7.23) you will use the keyboard.

(7.24) what will you use?

(7.25) who will use the keyboard?

(7.26) will you use the keyboard?

(7.27) you used the keyboard.

(7.28) what did you use?

(7.29) who used the keyboard?

(7.30) did you use the keyboard?

Not that examples 7.25, and 7.29, are used in language not only as questions, but also as relatives, as in *I saw the man who used the keyboard get on the bus.* Consequently, we might expect this example to cluster significantly differently from the others, but the others, being sentences, to be fairly similar to each other.

This is indeed the case. In the classification used to derive the categories listed above, we find the classification of the examples above in column A. If the classification is made slightly coarser, resulting in fewer **S**-level categories, then the classification of examples is as under column B.

Example sentence	A	B
you will use the keyboard.	S14	T6
what will you use?	S16	T6
who will use the keyboard?	S51	T21
will you use the keyboard?	S16	T6
you used the keyboard.	S14	T6
what did you use?	S16	T6
who used the keyboard?	N/A	T23
did you use the keyboard?	S16	T6

The category T6 corresponds to sentences, and sentences missing a compliment on the right (which is why it wasn't used in the examples above). As such, it is not a very linguistically coherent category, but nevertheless it captures the similarity between three of the four forms of sentence above, and as such is indicative that such similarities might be more coherently elucidated by more sophisticated versions of the techniques described here.

7.8 Discussion

This represents the limits of the work I have done in attempting to uncover linguistic structure in an entirely unsupervised manner from untagged linguistic data. All of the major linguistic units above the level of word have been partially uncovered — word classes, noun phrases, prepositional phrases, verb phrases, sentences. Consequently, I consider that this technique promises to uncover sufficient linguistic structure to serve as a 'bootstrapping' module in a machine syntax acquisition system.

There are many question concerned with what to do with the categories which have been derived. In the first place, since this technique reveals linguistic units which are strongly represented by simple statistical redundancies in the corpus, it is possible that this might serve to inspire statistical language models (after the work of Shannon, Jelinek, Schabes, and the like) which are likely to be highly effective in utilising the redundancy in corpora to predict future text given previous text. Firstly, it could be used to find a first estimate for the parameters of a context-free grammar or hidden markov models, since algorithms to find context-free grammars or hidden markov models are prone to finding local maxima, and are consequently sensitive to the starting point.

Secondly, this technique could be extended to find even higher level structure in much the same way that the structure in **C**-level sequences was extended to find structure in longer sequences.

Thirdly, it might be possible to symbolically analyse the categories derived using this method and provide a compressed rule-based description of them (see Wolff 1977) which can serve as a first approximation to a grammar of the corpus, which can be amended

by other procedures later.

Chapter 8

Neural Network Implementations

Neural Networks (henceforth, NNs), is a paradigm of computation largely inspired by the patterns of neurons and their inter-connections in the biological neural systems of animals. The analysis of computations neural networks perform differs from the analysis of computations most naturally described by a Turing Machine (TM) in the following ways:

- Whereas a TM computation is naturally defined by a function which is a semantic interpretation of a program (eg. Stoy, 1977), a neural network is most conveniently thought of as a device which generalises to a function from a sample of values of the function. That is, the goal of a neural network is typically to learn and generalise from examples it is given, while the goal of a TM is to compute a function from a specification of a function it is given in the form of a program.

- To facilitate analysis and definition of its operation, a neural network can naturally be considered as comprising two functions: a recall function which denotes the approximation the network has learned, and a training function which finds an appropriate recall function given a sample of values of the function to be learned.

- Descriptions of the *algorithms* which NNs use in both recall and training have natural descriptions involving the use of the following primitive terms: *units* (corresponding to neurons), *connections* (corresponding to the biological axions, dendrites and fibres linking neurons together in a network), *weights* (corresponding roughly to the biological concept of synaptic strength), and *networks* (corresponding to biological networks of neurons linked together by various fibres). Algorithmic descriptions of the functions Turing Machines can evaluate are naturally described by the relationship between the syntactically well-formed combination of various primitive operations (eg. commands, or logical deductions), and the order in which these are executed in performing a computation (or proof, if the definition of computation is interpreted logically).

One issue in the NN paradigm, therefore, is that of *neurobiological plausibility.* Do the training function, recall function and structure of the network have biological interpretations which accords with the empirical facts known about real neural networks? One problem here is that the biological evidence is, in many cases, equivocal, sometimes to the point of being contradictory. However, there are some principles which it is thought real neural networks possess which must be satisfied if biological plausibility is to be claimed.

Some of these principles are most naturally illustrated with reference to an example of a very simple model of a neuron: the perceptron (see Figure 8.1). The idea underlying the model is that the neuron varies its rate of firing depending on the current electrochemical activity in these fibres (and possibly the history of electrochemical activity in these fibres, and the chemical context of the neuron). In this simple model, the perceptron is a bi-state device, and the activities of its input fibres are also bi-state. That is, each input fibre can be thought of as being in one of two states — on or off, mathematically modelled as 0 or 1. Each synapse between the input fibre and the neuron is modelled here as a weight, which is interpreted as a real scalar.

The perceptron model says that the decision about whether the neuron should fire or not is made by a simply adding various pieces of evidence from the input fibres together in a weighted sum, and if this is above some threshold, the neuron fires; otherwise it

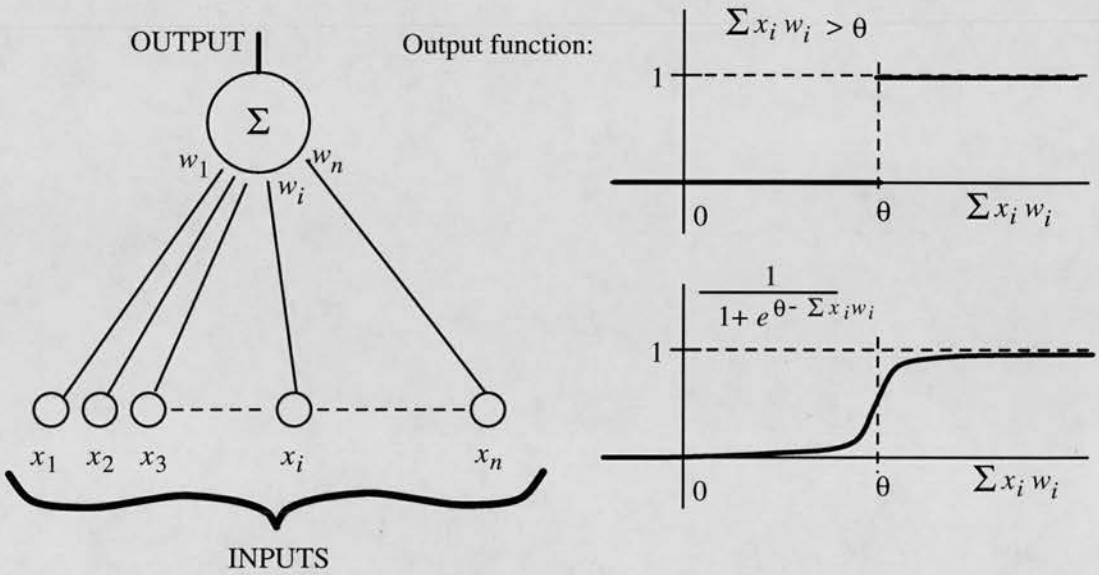


Figure 8.1: The perceptron model of the neuron. The n input fibres are modelled as having a certain amount of electrochemical activity, x_i. This is often taken as being either 1 or 0. At each synapse, there is learned weight, w_i. The perceptron's output is modelled as being monotonic in $\sum_{i=1}^{n} x_i w_i$, and two proposed functions are illustrated. The upper function, sometimes called a "cut-off" function, is 0 if $\sum_{i=1}^{n} x_i w_i \leq \theta$, and 1 if $\sum_{i=1}^{n} x_i w_i > \theta$, and a sigmoid function, $\frac{1}{1+e^{\theta - \sum_{i=1}^{n} x_i w_i}}$, which is a smooth version of the upper function.

doesn't. The way this is modelled is by assuming that the neuron fires if it receives more than a certain amount of electrochemical activation, and that each input fibre gives it activation in proportion to the synaptic strength between the fibre and the neuron if the input fibre is on, otherwise, it gives it no activation. Secondly, the assumption is made that the activation given by any set of fibres is the sum of their individual activations. Thus, the computation the simple perceptron model performs after it has been trained can be described mathematically as follows.

$$\text{Fire just in case } \sum_{i=1}^{N} w_i x_i > \theta$$

or

$$\text{Activation equals } \frac{1}{1 + e^{\theta - \sum_{i=1}^{n} x_i w_i}}$$

where θ is some real number threshold, w_i is a real valued synaptic strength, and x_i is a binary (0-1) valued electrochemical signal from fibre i.

In trivial ways, this is clearly not biologically plausible, or at least not biologically *faithful.* Real neurons fire in pulses, and cannot be faithfully modelled as being just either on or off, or even just summarised by a scalar firing rate. However, as an approximation, it is possible to argue that the on-off nature of the inputs reflects the fact that in certain areas of the brain, neurons either pulse very frequently, or not at all, so the value of the input fibre represents some short time average of the activity in the fibre, and this is idealised as a constant number. The unspoken assumption is that this idealisation is relatively irrelevant to the important aspects of the functioning of the network — the neuron is relatively functionally insensitive to the difference between a more faithful model which directly models the spiking, and this simplified model which ignores it. This may or may not be true[1], but the increased mathematical elegance of the simple model, and its relative ease of simulation, argue in favour of keeping it as an idealisation in order to model other conjectures about the functioning of the brain.

[1] In fact, the perceptron is not thought to be a very useful model of the functioning of individual neurons.

However, even with such an idealised model as the perceptron, interesting questions of biological plausibility can be addressed concerning, for instance, the availability of information needed by a neuron during the training process, and the nature of the training process itself.

Firstly, it is generally agreed that learning is a gradual, incremental, process — any change of brain state due to learning is a gradual, and in a sense constant, process. This rules out learning paradigms which 'batch' input, change the state of the network according to this input, and then when learning is finished, change the state of the network in some 'once only' way. For instance, in the algorithm for defining similarities between words given in chapter 5, bigram tables of words against contexts are collected, and *then* these are normalised to yield a new table from which similarities are calculated directly. This process of 'normalisation', if proposed as a neurobiological component of a training algorithm, would be deemed implausible because of the discontinuity between the collection of the data and the normalisation process.

Secondly, any information needed to update the value of a synapse due to training must be present *at the synapse being updated.* Consequently, it can make no reference to the values of other neurons or synapse values unless the value of the other neurons or synapse values can make themselves known at the synapse. This consideration has led many scholars to reject gradient descent based training algorithms, such as *backpropagation* (Rumelhart et al. 1986), as being neurobiologically plausible.

This chapter seeks to show how the processes of bigram data collection and normalisation may be 'folded' into each other so that there is no such discontinuity, and consequently the process of calculating similarity between words might be realised in a neural network. Secondly, it gives some results from actual simulations of one such network, demonstrating that a linguistically interesting clustering can be achieved.

The structure of the chapter will be as follows. First, a description of linear associative networks will be given in order to elucidate the similarity between these models of neural systems, and the statistical structures described earlier in the thesis. Secondly, it will be demonstrated how techniques first proposed by Minsky & Papert (1969) in relation to training rules for Bayesian classifiers can be applied to define a training algorithm

which finds a normalised contingency table incrementally. Finally, a network will be presented which uses topographic mappings to define a classification of lexical items syntactically.

8.1 Linear Associative Networks and Contingency Tables

Superficially, we shall see that there are striking similarities between Hebbian learning in an associative network, and the method described in chapter 5 of collecting bigram statistics. This similarity will now be explored.

Figure 8.2 shows a simple linear associative network, illustrating its relationship to the models used in this thesis. It has a certain number of input units, and a certain (other) number of output units. We shall be interested in determining how such a network might be used to learn a normalised bigram table of the type used in the experiments described in this thesis. Recall the definition of the bigram table $[\Psi, \Phi]$. According to its definition, cell $\langle \psi_i \phi_j \rangle$ of the table is defined to be the number of times $\langle \Psi, \Phi \rangle(n)$ equals $\langle \psi_i, \phi_j \rangle$, for all n for which the stream is defined.

The simple associative network may be used to collect the contingency table $[\Psi, \Phi]$ in the following way. Associate with each cue unit an element of the alphabet of Ψ, $\mathcal{A}_\Psi = \{\psi_1, \psi_2, \ldots, \psi_N\}$. Associate with each associant unit an element of the alphabet of Φ, $\mathcal{A}_\Phi = \{\phi_1, \phi_2, \ldots, \phi_N\}$.

Now, with each element of $\mathcal{A}_\Psi$, ψ_i, associate the vector, $\mathbf{e}_i$ which has all components 0 except the component corresponding to the unit associated with ψ_i. Do the same for the elements of the alphabet of Φ.

With each value of $\langle \Psi, \Phi \rangle$, $\langle \psi_i, \phi_j \rangle$, associate the cue-associant pattern pair $\langle \mathbf{e}_i, \mathbf{e}_j \rangle$. This gives a well defined mapping between the training stream, $\langle \Psi, \Phi \rangle$ and a training set of cue-associant vectors. Moreover, if training proceeds according to a hebbian learning rule, where a weight is updated by 1 just in case there is coincident pre and post synaptic activity (see figure 8.2), then it is clear that the weight at the synapse between the unit associated with ψ_i, and the unit associated with ϕ_j will be $[\Psi, \Phi](\langle \psi_i, \phi_j \rangle)$,

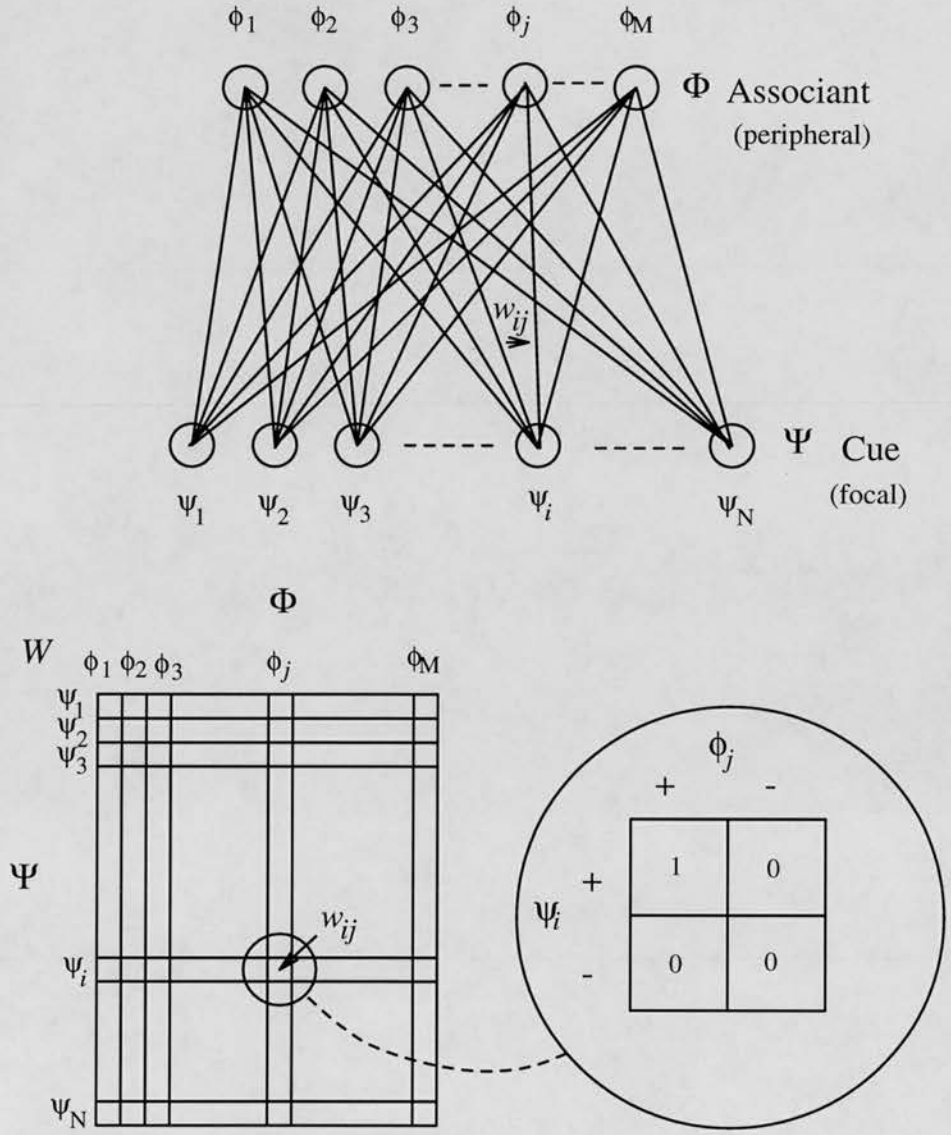


Figure 8.2: This figure shows a simple associative network (top). There are two sets of units — the Ψ ('cue') units, and the Φ ('associant') units. Between each pair of input-output units, there is a connection which ends in a synapse. Each synapse has a scalar weight associated with it, and this weight is learned by a training rule. The lower part of the figure shows the weight matrix, W, and the Hebbian training rule for the synapse between the units ψ_i and ϕ_j. If this training rule is used, and the stream $\langle \Psi, \Phi \rangle$ is used as training data, the values of synapse between ψ_i and ϕ_j will be $[\Psi, \Phi](\langle \psi_i, \phi_j \rangle)$.

since it is easy to see that coincident pre and post synaptic activity happens just in case the value of $\langle \Psi, \Phi \rangle$ is $\langle \psi_i, \phi_j \rangle$. The problem to be addressed is whether a learning rule can be found which generates a weight matrix either identical with, or usefully similar to, the *normalised* contingency tables used in the experiments reported in this thesis ($\overline{[\Psi, \Phi]}$). This is an interesting objective for two reasons: firstly if such a neurally plausible algorithm can be found, then the possibility exists of analysing real neural systems as performing non-parametric statistics similar to those described in this thesis as a means of uncovering structure. Secondly, the means by which this will be achieved is very similar to previous proposals by Minsky and Papert for training rules for Bayesian classifiers, and so it adds to a body of work on generalising, and statistically interpreting, training rules.

I shall show how such a table might be calculated incidentally where the goal of the neural system is to make predictions about the data, rather than to analyse structure explicitly.

8.2 Using Prediction for Classification

To return to the *bootstrapping problem*, recall that the problem is that rules and regularities are naturally defined in terms of categories which exist to describe the data, yet these categories are useful only in so much as they facilitate concise descriptions of the regularities which exist in the data. One of the functions of a sophisticated theory of a domain is to make predictions within that domain, and both neural networks and statistical models can be trained to make predictions, as described in chapter 2. This section seeks to describe the link between prediction, implicit learning, and classification.

Implicit learning is a means of learning some set of regularities incidentally, while actually aiming to learn something else. For instance, if the goal of a statistical system is to learn how to predict the next word given the current word, and if having done this the state of the statistical system can be analysed so as to uncover a categorisation of the domain, for instance by performing a hierarchical cluster analysis, then it might be said that the categorisation is *implicit* in the state of the statistical system after it has

been fitted to the observed natural language data.

Neural networks are predictive systems too, and the various parts of a trained network might be analysed in a similar way to how parameters in the statistical system described above to find structure in much the same way that cluster analysis was applied to a normalised contingency table in the previous two chapters. With this in mind, I shall go on to describe two statistical models which have natural interpretations as neural networks. The first is a simple Bayesian classifier, and the second is a feature/value predictor. In both cases, the analysis will be statistical, but weights can be interpreted as various functions of probabilities, and sigmoid functions of linear sums as probabilities.

8.3 Simple Prediction Models

The first model of data we shall consider is the same as that considered by Minsky & Papert (1969) in *Perceptrons*. The data from the positive and negative training examples are a set of vectors from a *product Bernoulli* random process. That is, a process where N random variables X_i are dependent on the value of a binary variable Y, such that X_i is independent of X_j for $i \neq j$ given the value of Y. That is, component i is 1 or 0, being 1 with probability p_i, $i = 1, 2, ..., N$. Thus we get a sample of vectors, each component of which is either 1 or 0, generated by a random process in which the values of the components are statistically independent.

The job of a predictive system is to infer the most likely value of Y given knowledge of the values of the $\{X_i\}$. Bayes' Theorem allows us to conclude that:

$$(8.1)\quad P(Y = 1|X = x)P(X = x) = P(X = x|Y = 1)P(Y = 1) = P(X = x\&Y = 1)$$

where X is a vector of random variables, $< X_1, X_2, ..., X_n >$, with a particular value $x =< x_1, x_2, ..., x_n >$

Let us first consider the situation where the variables $\{X_i\}$ are assumed independent if the training variable Y is 1. So by this assumption of conditional independence

$$(8.2)\quad P(Y = 1|X = x)P(X = x) = \prod_{i=1}^{n} P(X_i = x_i|Y = 1)P(Y = 1)$$

so

$$\log P(Y=1|X=x)+\log P(X=x)=\sum_{i=1}^{n}\log P(X_i=x_i|Y=1)+\log P(Y=1) \tag{8.3}$$

If X_i takes on values in $\{0,1\}$, we can write $P(X_i=x_i|Y=1)$ as,

$$\begin{aligned} P(X_i=0|Y=1)\left(\frac{P(X_i=1|Y=1)}{P(X_i=0|Y=1)}\right)^{x_i} &= \begin{cases} P(X_i=1|Y=1) & \text{if } x_i=1 \\ P(X_i=0|Y=1) & \text{if } x_i=0 \end{cases} \\ &= P(X_i=x_i|Y=1) \end{aligned} \tag{8.4}$$

So in this case, (8.3) can be rewritten as

$$\begin{aligned} \log P(Y=1|X=x)+\log P(X=x) &= \sum_{i=1}^{n}\log P(X_i=0|Y=1) \\ &\quad +\sum_{i=1}^{n} x_i \log\frac{P(X_i=1|Y=1)}{P(X_i=0|Y=1)} \\ &\quad +\log P(Y=1) \end{aligned} \tag{8.5}$$

where the first term on the right hand side in (8.3) is the sum of the first two terms in (8.5).

If we write

$$p_i=P(X_i=1|Y=1) \qquad 1-p_i=P(X_i=0|Y=1) \tag{8.6}$$

from (8.5) we get

$$\begin{aligned} \log P(Y=1|X=x)+\log P(X=x) &= \sum_{i=1}^{n}\log(1-p_i) \\ &\quad +\sum_{i=1}^{n} x_i \log\frac{p_i}{1-p_i} \\ &\quad +\log P(Y=1) \\ &= C+\sum_{i=1}^{n} x_i w_i \end{aligned} \tag{8.7}$$

Where C does not depend on x.

Now, (8.7) is of the form calculable by a linear (perceptron-style) unit. Moreover, the exponent of its value is clearly $P(Y = 1 \wedge X = x)$. Also, any set of *weights* of a linear unit may be interpreted probabilistically in this model: We simply note that since

$$w_i = \log \frac{p_i}{1 - p_i} \tag{8.8}$$

Then

$$p_i = \frac{e^{w_i}}{1 + e^{w_i}} \tag{8.9}$$

In terms of the perceptron model of figure 8.1, the weights w_i are as described in 8.8, and the exponential of the linear sum is $P(Y = 1 \wedge X = x)$.

8.3.1 Likelihood Ratio

Now, if the negative examples too are assumed to be generated from a product Bernoulli source, then the analysis can be further extended to compute the probability *ratio*

$$\frac{P(Y = 1|X = x)}{P(Y = 0|X = x)} \tag{8.10}$$

This is interpreted as the *odds on* $Y = 1$ given the value of the vector X.

Algebraic manipulation of (8.10) lead to concluding that this quantity is

$$\frac{\prod_{i=1}^{n} P(X_i = x_i|Y = 1)}{\prod_{i=1}^{n} P(X_i = x_i|Y = 0)} \times \frac{P(Y = 1)}{P(Y = 0)} \tag{8.11}$$

and taking logs of (8.11), and taking note of the identity (8.4), we derive that

$$\begin{aligned} \log \frac{P(Y = 1|X = x)}{P(Y = 0|X = x)} \quad = \quad & \sum_{i=1}^{n} \log \frac{P(X_i = 0|Y = 1)}{P(X_i = 0|Y = 0)} \\ & + \sum_{i=1}^{n} x_i \log \frac{P(X_i = 1|Y = 1)}{P(X_i = 0|Y = 1)} \frac{P(X_i = 0|Y = 0)}{P(X_i = 1|Y = 0)} \\ & + \log P(Y = 1) - \log P(Y = 0) \end{aligned} \tag{8.12}$$

and writing p_i for $P(X_i = 1|Y = 1)$, and q_i for $P(X_i = 1|Y = 0)$, we derive

$$\begin{aligned} \log \frac{P(Y=1|X=x)}{P(Y=0|X=x)} &= \sum_{i=1}^{n} \log(1-p_i) - \log(1-q_i) \\ &\quad + \sum_{i=1}^{n} x_i \log \frac{p_i}{1-p_i} \frac{1-q_i}{q_i} \\ &\quad + \log P(Y=1) - \log P(Y=0) \end{aligned} \tag{8.13}$$

which is again of the form

$$\log \frac{P(Y=1|X=x)}{P(Y=0|X=x)} = \sum_{i=1}^{n} w_i x_i + C \tag{8.14}$$

so it is readily calculable by a linear unit. Moreover, in this case, since $P(Y = 1|X = x) = 1 - P(Y = 0|X = x)$, we observe that the standard sigmoid function, $\frac{1}{1+e^{-x}}$, maps (8.14) to $P(Y = 1|X = x)$. So, the standard non-linear transformation applied to the value calculated by a linear unit has a simple probabilistic interpretation. This has been mentioned by others, including Hinton (1989) and Spackman (1992), and can be generalised to multi-layer networks.

In this interpretation, we note that the weights have an interpretation in the parameters of the product Bernoulli process as:

$$w_i = \log \frac{p_i}{1-p_i} \frac{1-q_i}{q_i} \tag{8.15}$$

and the constant, C, has interpretation

$$\log P(Y=1) - \log P(Y=0) + \sum_{i=1}^{n} \log(1-p_i) - \log(1-q_i) \tag{8.16}$$

Thus the standard sigmoid function applied to the linear sum of the perceptron in figure 8.1 yields the probability that $Y = 1$ given knowledge of the value of X, under the (statistical) assumption that X_i is independent of X_j, $i \neq j$ given knowledge of the value of Y.

8.3.2 Incremental Learning

We now turn to the question of how to estimate the value of the weights of (8.8) locally, and incrementally. It is an interesting question, particularly for the hypothesis that a system such as this might be neurally implementable — a learning rule which depends solely on the values of the current pre- and post- synaptic values is considered more likely to be neurally implementable than one which depends on more complicated relations within the training data. This section provides update rules for weights which are locally implementable if the neuron has a means to estimate the reciprocal of the time it has spent learning until now (or the number of training events). Update of the weights will happen only in the case of post-synaptic activity.

First, we must assume a training model. The training model we assume is now presented, motivated largely through mathematical perspicuity, though a number of more neurologically plausible variants will give rise to essentially similar analysis (eg. assuming that the variables are poisson processes).

The Training Model

We assume that the activity of each variable $\{X_i\}$ and Y in the model 8.3 is a function of time, being at each instant either 0 (low) or 1 (high). We shall consider just one of the weights, w_i.

For the purposes of this analysis, we shall write $X_i(t)$ for the value of X_i at time t. We shall define the probability of some predicate, $\phi(t)$ of the variables $\{X_i(t), Y(t) | i = 1, 2, ..., n\}$, up to time T, to be

$$(8.17)\ \frac{1}{T}\int_0^T \chi_{\phi(t)} dt$$

that is, the proportion of time ϕ has been true up until time T. It is clear that this model subsumes a model which assumes a discrete number of trials, since this can be modelled by splitting the time axis into units, and stating that each variable is either on or off for a unit of time.

Analysis

First, we estimate p_i, the probability of X_i given Y (see (8.6)). Now by definition,

$$(8.18)\ \delta p_i(t) = p_i(t+\delta t) - p_i(t) = \frac{\frac{1}{t+\delta t}\int_0^{t+\delta t} X_i(x) \wedge Y(x)dx}{\frac{1}{t+\delta t}\int_0^{t+\delta t} Y(x)dx} - \frac{\frac{1}{t}\int_0^{t} X_i(x) \wedge Y(x)dx}{\frac{1}{t}\int_0^{t} Y(x)dx}$$

Let us define the probabilities that Y and both X_i and Y are high estimated at time t by:

$$(8.19)\ \alpha(t) = \frac{1}{t}\int_0^t Y(x)dx \qquad \beta(t) = \frac{1}{t}\int_0^t X_i(x) \wedge Y(x)dx$$

From (8.18) & (8.19) we infer

$$(8.20)\ \delta p_i(t) = \frac{\beta(t+\delta t)}{\alpha(t+\delta t)} - \frac{\beta(t)}{\alpha(t)}$$

If $Y(x) = 0; x \in [t, t+\delta t]$, then it is easy to verify that $\delta p_i(t) = 0$. Otherwise, there are two cases.

Case: $X_i(\tau) = 0, \tau \in (t, t+\delta t)$. First, we note that

$$(8.21)\quad \begin{aligned} \beta(t+\delta t) &= \tfrac{t}{t+\delta t}\ \left(\tfrac{1}{t}\int_0^t X_i(x) \wedge Y(x)dx + \tfrac{1}{t}\int_t^{t+\delta t} X_i(x) \wedge Y(x)dx\right) \\ &= \tfrac{t}{t+\delta t}\ \left(\tfrac{1}{t}\int_0^t X_i(x) \wedge Y(x)dx + \quad 0 \quad\right) \\ &= \tfrac{t}{t+\delta t}\ \beta(t) \end{aligned}$$

Similarly,

$$(8.22)\ \alpha(t+\delta t) = \frac{t}{t+\delta t}\left(\alpha(t) + \frac{\delta t}{t}\right)$$

From (8.20), (8.21), and (8.22) we see that

$$(8.23)\ \delta p_i(t) = \frac{t\beta(t)}{t\alpha(t) + \delta t} - \frac{\beta(t)}{\alpha(t)}$$

By the Taylor expansion for $(1+x)^{-1}$, we see that

$$(8.24)\ (t\alpha(t) + \delta t)^{-1} = \frac{1}{t\alpha(t)}\left(1 - \frac{\delta t}{t\alpha(t)} + o\left(\frac{\delta t}{t\alpha(t)}\right)^2\right)$$

By substituting (8.24) in (8.23), it is easy to verify that

$$(8.25)\ \delta p_i(t) \approx -\frac{\delta t}{t\alpha(t)} p_i(t)$$

this being true in the limit as $\delta t \to 0$.

Case: $X_i(t) = 1$. From (8.20), and a similar analysis to that presented in (8.21) and (8.22) we see that

$$(8.26)\ \delta p_i(t) = \frac{t\beta(t) + \delta t}{t\alpha(t) + \delta t} - \frac{\beta(t)}{\alpha(t)}$$

Again, by substituting (8.24) into (8.26), we find that

$$(8.27)\ \delta p_i(t) \approx \frac{\delta t}{t\alpha(t)}(1 - p_i(t))$$

So this analysis shows how p_i varies with its training data over time. What we want is *weight* update rules. The weights are related to the conditional probabilities by (8.8), and by applying the Chain Rule for differentiating functions of functions, we find that

$$(8.28)\ \frac{dw(t)}{dt} = \frac{d \log \frac{p(t)}{1-p(t)}}{dt} = \frac{1}{p(t)(1-p(t))} \frac{dp(t)}{dt}$$

From (8.28) and (8.25), we see that if $X_i(t) = 0$, then

$$(8.29)\ \delta w \approx -(1 + e^w)\frac{\delta t}{t\alpha(t)}$$

and from (8.28) and (8.27), then if $X_i(t) = 1$, then

$$(8.30)\ \delta w \approx (1 + e^{-w})\frac{\delta t}{t\alpha(t)}$$

Finally, if $Y(t) = 0$, then from (8.28), and the observation that $\frac{dp(t)}{dt} = 0$, we see that

$$(8.31)\ \delta w = 0$$

Consequently, we can construct an update table for weights as follows

(8.32)

δw_i	$Y(t)=1$	$Y(t)=0$
$X_i(t)=1$	$(1+e^{-w_i})\frac{\delta t}{t\alpha(t)}$	0
$X_i(t)=0$	$-(1+e^{w_i})\frac{\delta t}{t\alpha(t)}$	0

Also, it is possible to find an update rule for $\alpha(t)$ by similar, though simpler, methods, noting that

$$(8.33)\quad \alpha(t+\delta t) = \frac{t\alpha(t) + Y(t)\delta t}{t+\delta t}$$

to yield

$$(8.34)\quad \frac{d\alpha(t)}{dt} = \begin{cases} \frac{(1-\alpha(t))}{t} & \text{if } Y(t)=1 \\ -\frac{\alpha(t)}{t} & \text{if } Y(t)=0 \end{cases}$$

Consequently,

(8.35)

	$Y(t)=1$	$Y(t)=0$
$\delta\alpha$	$\frac{(1-\alpha(t))\delta t}{t}$	$-\frac{\alpha(t)\delta t}{t}$

Now, by keeping track of $\alpha(t)$, and $w_i(t)$, the linear unit can calculate the log of the probability that the vector $< x_i >$ is on its inputs while its training variable is 1, since

$$(8.36)\quad \log P(X = x \& Y = 1) = \log\alpha + \sum_{i=1}^{n} \log\frac{1}{1+e^{w_i}} + \sum_{i=1}^{n} x_i w_i$$

This approach can be used to derive incremental rules for deriving many functions of probabilities. Derivation of the parameters for the likelihood ratio model will not be given here for reasons of space and relevance, but it is clear that the derivation follows similar lines.

8.4 The Feature/Value (Product Multinomial) Model

Until now, we have assumed that the data from which prediction is to be made are from a product Bernoulli process. In this model, each of the n X_i take one of two values. Let

us generalise this model so that each X_i takes one of M_i values. This model has relevance to the commonly used "feature/value" representational scheme in connectionist systems. In this representational scheme, an input vector is arbitrarily split up into a number of sub-vectors, each sub-vector corresponding to a certain "feature". Each sub-vector is associated with a number of components, each component corresponding to the *value* of that feature. The values of each feature are mutually disjoint. For instance, the "features" could be *previous word* and *next word*, and the values of these features would be the identity of the previous word, and the identity of the next word. This model clearly has relevance to the classification techniques used in this thesis.

In this model, we have a number of features X_i, taking values $x_i \in F_i$, being used to predict the value of another feature Y taking values in G, with the following statistical assumption of independence:

Assumption of Feature Independence: X_i is independent of X_j, $i \neq j$, given the value of Y.

To interpret this as a neural network, consider the associative network of figure 8.2. The lower units are split into 'banks' of 'features', the nodes in each feature being exclusive and exhaustive for any data item (any cue vector will have precisely one input in each 'feature' high). The nodes on the upper layer correspond to the possible values of Y (which are now assumed to be more than two).

The statistical analysis of this model is similar to the analysis of the product Bernoulli model of section 8.3 above. In fact, it is clearly possible to derive an analogous equation to (8.3) above, namely:

$$\log P(Y = y|X = x) + \log P(X = x) = \sum_{i=1}^{n} \log P(X_i = x_i|Y = y) + \log P(Y = y) \tag{8.37}$$

Now, since for each feature X, we have, in this model, that

$$\sum_{j=1}^{M_i} P(X_i = f_j|Y = y) = 1$$

the situation is rather less complicated that that of the product Bernoulli model. In fact, we can rewrite (8.37) as

$$\log P(Y = y|X = x) + \log P(X = x) = \sum_{i=1}^{n} \sum_{j=1}^{M_i} x_{ij} \log P(X_i = f_j|Y = y) + \log P(Y = y) \tag{8.38}$$

where x_{ij} is the component of X_i associated with the jth feature, the double sum simply being the sum over all the entire vector X. It is clear that (8.38) is again readily computable by a linear unit.

Thus, to be able to predict Y from X, it is necessary in this model to calculate the log of the conditional probabilities $P(X_i = f_j|Y = y)$. These values can be calculated incrementally in much the same way that the coefficients in the product Bernoulli process were shown to be incrementally calculable above, in fact, incremental rules for updating the conditional probabilities were derived in (8.27) and (8.25) above, so incremental rules for updating the logarithm of these can be found by applying the chain rule in much the same way as was done in finding update rules for $\log \frac{p}{1-p}$ above.

8.5 Incremental learning and contingency tables

Recall that in the analysis of natural language corpora described earlier in this thesis, an analysis was made of a normalised contingency table, $\overline{[\Psi, F(\Psi)]}$. This section shows how the incremental learning rule derived above can be used to calculate this contingency table 'on line'. Let us first recall what the definition of normalisation was for the contingency table, as defined in equation 5.4. It was defined as

$$\overline{[\Psi, F(\Psi)]}_{\lambda\mu} = \Omega_E([\Psi, F(\Psi)])_{\lambda\mu} = \frac{[\Psi, F(\Psi)]_{\lambda\mu}[\Psi, F(\Psi)]_{++}}{[\Psi, F(\Psi)]_{\lambda+}[\Psi, F(\Psi)]_{+\mu}}$$

Now, from elementary probability theory, we have

$$P(\Psi = \lambda|F(\Psi) = \mu) = \frac{P(\Psi = \lambda \& F(\Psi) = \mu)}{P(F(\Psi) = \mu)} = \Omega_E([\Psi, F(\Psi)])_{\lambda\mu} \frac{[\Psi, F(\Psi)]_{\lambda+}}{[\Psi, F(\Psi)]_{++}} \tag{8.39}$$

Now we know that in performing prediction of context in the 'feature/value' model using linear units, one needs to calculate the coefficients $P(\Psi = \lambda|F(\Psi) = \mu)$, or

$\log P(\Psi = \lambda | F(\Psi) = \mu)$, so these values will be available from an attempt to fit a simple prediction model to natural language data. However, if a measure based solely on ranks is used to measure distances between focal items, such as the SRCC used earlier, then consider the vector of normalised values, $\Omega_E([\Psi, F(\Psi)])_{\lambda X}$, where X ranges over the alphabet of $F(\Psi)$ (thus this is the vector associated with the focal item λ). If this vector is multiplied by a constant, such as it is in (8.39) to calculate $P(\Psi = \lambda | F(\Psi) = \mu)$, then the rank ordering of the values of the alphabet of $F(\Psi)$ *remains unchanged.* This implies that the distances calculated between items will remain unchanged too. More generally, this is also the case if all the components of the vector are operated on by any monotonic increasing function (eg. $\frac{x}{1-x} : x \in [0, 1)$, or logarithm for $x > 0$).

Consequently, if the conditional probabilities $P(\Psi = \lambda | F(\Psi) = \mu)$ can be incrementally calculated (as they can[2]) for solving a prediction problem, then direct application of rank-based distance measures is possible to derive topological structure within the representational space. One way this may be achieved is now investigated.

8.6 Topographic Mappings

I have shown that the parameters necessary for the direct application of simple metrics to uncover structure occur naturally in probabilistic models of prediction which might be implemented in NN hardware such as perceptrons and associative networks, and that training rules exist which update the values of these parameters incrementally, and hence are not immediately implausible neurological models[3]. I have not shown how such a network might be analysed so as to find classifications of natural language. This section does just this.

As described in chapter 5, the approach to uncovering linguistic structure pursued in this thesis has two components — a means of finding the local similarity structure of the representational domain, and a means of extracting structure from this similarity

[2] As discussed above, these parameters were shown to be incrementally calculable in (8.27) and (8.25). Since logarithm is a monotonic function, calculating $\log P(\Psi = \lambda | F(\Psi) = \mu)$ will suffice as a set of parameters suitable for the direct application of the SRCC. These parameters occur naturally in Bayesian models of prediction such as those described above.

[3] Although probably not, as they stand, particularly neurologically plausible either.

data. There are many ways to extract structure, and the one used in this thesis so far is a clustering algorithm after the work of Sokal & Sneath (1963). However, there are also connectionist algorithms which can be used to extract structure in this way, under the general heading of 'self organising systems'. Two such algorithms are due to Kohonen (1982), and Willshaw & Durbin (1987).

The idea in both of these algorithms is to define a topology over a space we wish to map our data into, and then use a connectionist algorithm to find that mapping which preserves as much of the similarity data as possible.

The network I used for the simulations reported here is based on one due to Kohonen (1982). Work has been done using these methods by Scholtes (1991a,b), but he didn't have the advantage of a prior statistical analysis of the problem, and consequently his results are less impressive. The goal of the system is to produce relatively coherent word classes. If the priorly defined topology is the discrete topology, the Kohonen network implements a variant of k-means clustering, where the k output units (or more exactly their weight vectors) correspond to the k-means which compete to account for portions of the data to which they are most similar. In this case, the k clusters found by this procedure will only be of interest if they correspond to linguistically perspicuous categories. This will depend on the way in which words are represented in the system.

Linguistic theory suggests that a good indicator of the syntactic category of an item is the distribution of linguistic contexts in which it can occur. To represent context without assuming any prior knowledge of linguistic structure, words are represented simply by their normalised bigram statistics, these being learned by training between layers 1 and 2 in the network of figure 8.3. These statistics can be collected by a learning mechanism such as that described above, and then passed to a simple Kohonen network.

The network shown corresponds to that used in the simulations with large corpora of real text. A similar, scaled down, version was used in the letter and phoneme level simulations reported below. The network can best be thought of as comprising two components — the lower two layers are used to train the network with natural language data using the algorithms described above to learn a normalised representation of the data, and the upper two layers are a simple Kohonen network which generates a

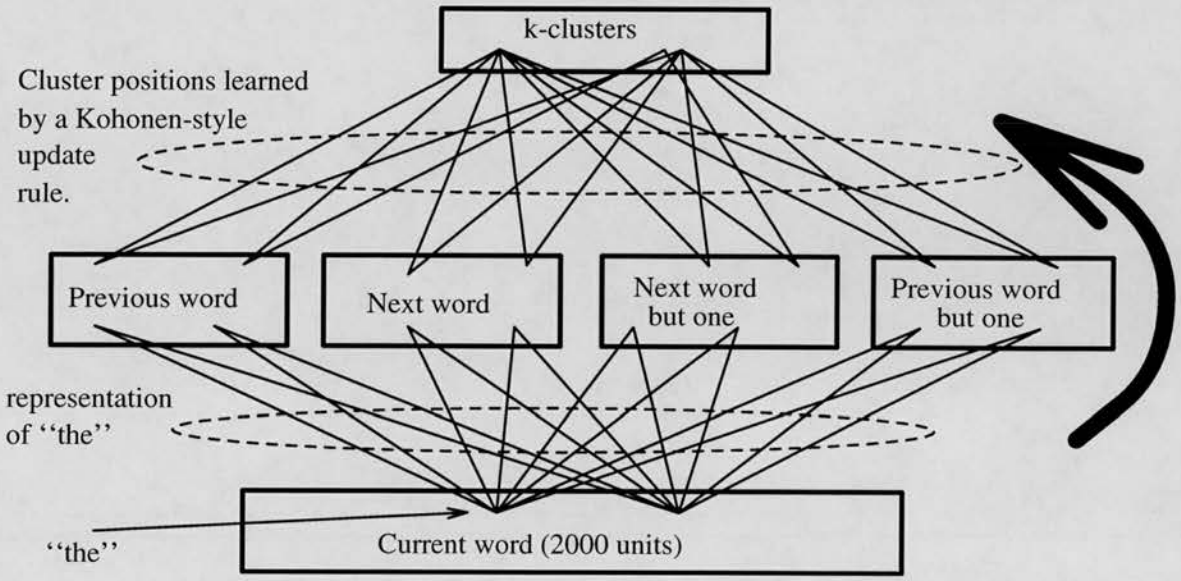


Figure 8.3: This is the network used to induce topological structure. Training initially proceeds between layers 1 & 2, the focal word being used to predict the values of the contextual "features". In so doing, the network effectively computes a version of the normalised contingency table incrementally, as described above in section 8.4. This representation is then exploited by the Kohonen network to calculate a version of k-means clustering, which imposes a topological structure over the focal items.

topographic mapping between the representations (on the middle layer) and a set of categories (on the top layer). This combined process induces a categorisation of words by defining two words to be in the same category just in case their representations are mapped to the same category by the Kohonen network.

The lower set of units use a localist representation of the current word (there are 2000 units, each corresponding to a different word under study). The middle set of units are divided into 4 banks, one bank corresponding to each of the four contextual bigram relations considered: last word but one, previous word, next word, next word but one. Only the most common 150 words were considered, and appearances of all other words in these contextual relations were ignored. The net was trained using a corpus of 40,000,000 words from USENET newsgroups using simple Hebbian learning, with normalisation, as described above. After training, if a current word is presented, the middle layer will represent the distribution of contexts in which that word occurs. Once the first layer is trained, the learnt distributions of each of the 2000 words are clustered into 100 groups using the Kohonen part of the network.

In order to define a Kohonen-style network, we generally need to make three design decisions. These are discussed here.

1. The topology of the top layer — how updating the weights of one of the units on the top layer leads to the updating of the weights of the other units on the top layer. This is concerned with the topology we wish to assume exists in the data. If the topology is assumed to be continuously mappable to a section of the Euclidean plane, then it may be desirable to give this a grid structure (eg. Kohonen, 1982; Durbin & Willshaw 1987). Since the goal of this process is just to induce *categories*, there is no need to assume any structure at this level at all, so updating any one of the top units will have no effect on any of the others.

2. How a new example representation on the middle layer updates the weights of units on the top layer. In the vanilla Kohonen network, there is a 'winner takes all' approach to this problem. A 'similarity' metric is defined between the weights attaching to a unit on the top level, and representations on the middle layer. The unit whose weights are closest to the new piece of data is updated so that

its weights become still closer to the new piece of data than before. Then the units in the top layer are updated in accordance with the assumed topology as described above. The definition of this 'similarity metric' is usually some variant of Euclidean distance ($\mathbf{L}_2$), but in line with the discussion in chapter 5 on the statistical utility of various similarity metrics for the purposes of classification, and with the good results which were obtained by rank correlation coefficient, this is the metric which will be used in this version of the Kohonen network. This update function also often changes with time — early presentations cause a relatively large change in the weights of the top layer, while this change is reduced as time goes on.

3. The example presentation paradigm. There are two natural choices for how examples are presented for clustering by the network. The first is to simply present the 2000 words we are considering one at a time, and cycle this presentation routine several times. The second is to present words as they appear in a corpus. This latter presentation paradigm is both more natural (since examples are presented as they are 'seen'), and leads to better results, so that is the paradigm that is used here. It might be more natural still to 'fold' the processes of learning the representation (between layers 1 and 2), and learning the categorisation (between layers 2 and 3) by learning and clustering at the same time. Although this is natural, this was not done here because to do so would be computationally intractable with the machinery available (since each word in the corpus would be associated with too many similarity calculations. This would present no difficulties for a highly parallel system).

8.6.1 Network Simulations

First, a small network was given the task of clustering together letters. The lower two layers of the network were trained using the training algorithm described above. Thus, the focal letter was represented in the lowest layer of the network, and the four surrounding letters were represented in the four banks of units in the middle layer. Thus, when training was complete, the set of weights between the lowest layer and the middle

layer were the same normalised weights as were obtained in the letter-level experiment of section 6.2.4[4].

Kohonen clustering was then performed by presenting letters from a corpus to the lowest layer one by one, setting the value of a node in the middle layer to be the value of the weight between it and the unit representing the current focal letter in the lowest layer, and using the entire middle layer as input to the Kohonen clustering process. When the upper layer of the network consisted of two cluster nodes, the induced mapping between letters and clusters was found to correspond precisely to the division between vowels and consonants.

In the word-level experiments, training between layers 1 and 2 could have proceeded in the same way as for the orthographic experiment described above. However, this would have been time consuming, and advantage was taken of the fact that the weights between layers 1 and 2 can be mapped onto cells in the normalised contingency table of the word level experiments of chapter 6, and this contingency table was used directly as the weight matrix between layers 1 and 2.

The upper layer of the network was initially assigned 100 nodes with weights assigned so that each of the 100 most common words had a node in layer 3 assigned with weights so that the distance between the word and the node was 0. 10,000 words were then taken from newsgroup articles and presented to the network which proceeded to cluster the presented tokens. In general, words in the same cluster tend to have the same syntactic category, although there is sometimes more than one cluster which corresponds to the same syntactic category. Also some clusters appear to correspond to no linguistic category. Some of the clusters are shown below. Notice that one of clusters corresponds not to a single linguistic category, but consists of words which are ambiguous between two linguistic categories: nouns and verbs. In many of the categories there are one or two apparently spurious items, and some of the smaller categories, not shown, do not appear have any coherent linguistic basis.

VERB: wish want understand trust tell suggest see saying say respect remember regard recognize realize prove notice mean let know knew include imagine hope hear forget figure feel fail expect

[4]That is, each weight corresponds to a cell of the normalised contingency table of that section.

except determine deny demand decide consider claim blame believe assume ask argue appreciate agree

VERB: work wear watch waste wait try throw thank tend teach take speak shoot serve sell require recommend pull provide produce present move make listen like leave learn join have hate happen gives give get find enjoy draw do disagree die compare choose catch carry buy begin became beat bear be avoid allow

PT-V: went treated supported started shot returned responsible removed referred received produced printed pointed picked needed lived listed killed involved intended installed gone going given gave developed defined created covered considered changed caused came built been applied along

PPL: used told thinks suggested stuck stated spent sold shown seen saw realized published provided proven proposed presented posted played placed noticed moved met mentioned learned informed included heard gotten found expressed experienced done discussed discovered died described convinced considering claimed checked caught brings aware assumed asked appeared allows

ING: watching trying telling taking starting showing sending selling seeing referring putting pointing paying moving making looking keeping having giving getting finding doing cross changing calling buying being asking

ADJ: wrong well valid useful sufficient successful sorry smaller silly safe ready rather ok obvious more longer impossible fast effective easier down careful better available accurate acceptable

NOUN: woman wife utility tree topic team surface style ship reason readers product practice people man machine line law information guy group game food family equipment company code church chance bomb background album

PL-N words women whom weapons versions users types tools teams stories songs sites references programs products parents others opinions movies men groups functions fonts features fans examples elements disks discussions computers companies classes cases articles arguments and

PRON: you we this they there someone somebody she saddam paul none nobody let's jim it israel immediately i he everything everyone everybody bush both anyone anybody americans

ADV: usually truly thus therefore then surely suddenly still sometimes somehow simply really probably possibly personally often obviously not normally never neither necessarily merely maybe likely just indeed hardly generally ever eventually essentially definitely currently clearly certainly basically apparently always also already actually

AMBIG:N/V: update transfer test split show set ride return release post log load install force focus feed face end echo cover copy charge

AUXIL: you're you'll you'd wouldn't would won't will we've we'll shouldn't should shall seems must might may isn't i've i'll i'd haven't hadn't had gonna finally dont don't doesn't does didn't did couldn't could cannot can't can

CONJ: yet why while whether where when whatever what until unless though that so since now least if how here but before because as again

PREP: without within with via using upon under towards toward to through regarding over outside on of near into inside including includes in from for during contains by beyond between behind becomes at among against after across about

8.6.2 Benchmarking network performance

Although the network does fairly well, it is by no means finding all the linguistic structure implicit in the bigram statistics. This can be seen by using a conventional hierarchical cluster analysis of these statistics, which clusters almost all the data into 30 main categories, corresponding to major linguistic categories, and in many cases, fine-grained semantic factors are revealed within these categories (see Figure 2). In comparison, the network does not produce coherent categories with just 30 clusters, and with 100 clusters, there is only limited semantic similarity within clusters.

The empirical efficiency of this clustering technique (defined in section 6.3.2), when tested on the SUSANNE corpus as described in 6.3.2 was 64%, as opposed to 70% for the empirically derived classification of words given in chapter 6, so it does quite well.

8.7 Discussion

This chapter gives qualitative reasons to believe that the paradigm described in this thesis is neurally implementable, and that standard network techniques can be used to find categorical structure in natural language as described above. The efficacy of the approach is enhanced by taking statistical analysis of *what* is being done into account, and using statistical theory to motivate a learning rule and a similarity measure which

enhances the network's ability to uncover linguistic structure beyond using standard learning rules and similarity measures. Further, analysis was presented which demonstrated that, in particular, the normalised contingency table, shown to be efficacious in the previous two chapters for enabling the SRCC to uncover structure, could be calculated as a by-product of a simple prediction task.

A question remains as to whether the network presented here could be called neurally plausible, since it seems unlikely that rank correlation coefficient is neurally plausible, and no-one has suggested a way that the 'winner takes all' nature of the Kohonen network can be implemented neurobiologically. This latter problem can be overcome by using the *elastic net* of Durbin & Willshaw, which does not have a winner takes all nature, yet produces better topographic mappings (in some domains), and perhaps there are neurally implementable functions which are statistically closer to rank correlation coefficient than linear correlation coefficient or Euclidean distance. Nevertheless, the network presented here clearly uncovers significant structure at the word level, and could be modified (by making a unit on level 1 correspond to a sequence of previously uncovered categories) to cluster sequences together as described in chapter 7.

Chapter 9

Conclusion

I set out to show how a largely knowledge-free, bottom-up learning technique could be applid to natural language data to uncover significant amounts of linguistic structure without requiring a large quantity of innate knowledge. First of all, I shall summarise the technique, then I shall summarise the results, then I shall discuss some of the arguments in favour of nativism with respect to the results of the experiments of this thesis. Finally, I shall discuss some future directions I am currently investigating with respect to pursuing this work.

9.1 Review of the Classification Finding Technique

Firstly, punctately represent the individuals in the domain in which we wish to find structure, the *focal domain*. Find an informative relationship with another, statistically dependent, domain (the *peripheral domain*). The information in this relationship between the focal and peripheral domains will be exploited to find the structure of the focal domain, so this informative relationship defines the nature of the structure which will be uncovered. In natural language, adjacent words are mainly informative about the syntactic category of each other, so this is the relationship which is used to uncover syntactic structure. The relationship between fairly distant words (up to about 500 words away) is informative about the meaning of the words in this relationship, and

Brown et al. (1990) used this relationship to uncover a semantic classification. The relationship between the words written in an article, and what the written article is broadly about is also informative of the meaning of the words, and I used this relationship to uncover semantic classes.

The way in which the structure is uncovered is to some extent arbitrary — different techniques will work better in different domains dependent on the nature of the statistical redundancy which exists in the informative relationship being exploited. However, a successful technique for natural language data is to define a distance metric between elements of the domain whose structure we wish to uncover which measures the similarity of the statistical distributions, so that items in the focal domain which make similar predictions according to the informative relationship are judged as similar. In fact, the only hard and fast rule is that items which make statistically the *same* predictions should be judged as similar. Although measurements between distributions abound, the Spearman rank correlation coefficient was found to be particularly good at uncovering structure in language, and there are good statistical reasons to prefer it when the statistical nature of the regularity between the two domains is unknown.

Once a measure of similarity has been chosen, the similarities are used in a vanilla hierarchical cluster analysis, after Sokal & Sneath, to find a hierarchical classification of the focal domain.

Alternatively, a prediction problem can be defined from the informational regularity (predict the peripheral value given the focal value or vice-versa), and a statistical or neural network system trained with respect to the corpus to perform prediction. Once this has been done, standard neural network or statistical topographic mapping techniques can be applied to uncover a topology for the focal domain.

Thus I have described a rather general approach to finding structure in representational domains which is not in principle restricted to uncovering the structure of language, and certainly not to uncovering such structure from bigram statistics of the corpus, even though such statistics have been shown efficacious in uncovering such structure.

9.1.1 Review of the Experimental Results

Structure has been uncovered at many levels in natural language. When the informative relationship comprised *previous item, next item, last item but one*, and *next item but one*, at the level of the character and the phoneme, a clear distinction was found between vowels and consonants. At the level of the word, a hierarchical ontology was derived which showed clear distinctions between all the main syntactic categories, and which also appeared to make some finer grained semantic distinctions. When the relationship was between the words in a news article, and the name of the newsgroup in which it appeared, a classification was found which embodied certain semantic and morphological relationships between words, such as having a common stem, being associated with each other, and so on.

At higher levels, phrasal structure was found in natural language. Categorised sequences of length 1, 2, and 3 were clustered, and short noun phrases and prepositional phrases were evident, together with clusters corresponding to various verbal forms. When sequences of *these* categories were clustered, simple sentences and verb phrases also became evident, together with more complicated noun phrases and prepositional phrases.

9.2 Relation to Linguistics and Nativism

Although considerable structure was found in natural language using these techniques, it still seems somewhat unlikely that the subtle structure which modern linguistic theories would suggest needs to be extracted, can be extracted by these methods alone.

Where then does the nativist argument about knowledge of linguistic structure stand? The nativist argument was proposed by Chomsky (1965; 1980) as a means of giving a psychological interpretation to his theory of transformational generative grammar. It also relates somewhat to the efforts of the structuralists after Bloomfield, who sought to *define* language using bottom-up methods. Since the techniques presented here are bottom-up too, it will be interesting to discuss the relation of these methods to those used by the structuralists.

The structuralists sought to define all linguistic units, from the phoneme to the sentence, using a scientific empirical methodology. The methodology involved was rather procedural in nature, examples including the 'frame and focus' method to define the class of possible nouns as those words which could appear in a certain position in a sentence. For instance, consider the sentence *The ? is good.* The '?' can be replaced by some words but not others if the sequence is to be a sentence. For instance, '?' might be *car, man* or *idea*, but not *think, is* or *to.* Much more sophisticated versions of this approach can be used to define a set of classes, and the structuralist techniques of *immediate constituent analysis* can be applied to give classifications to phrases, and so on. Chomsky argued that this enterprise was far too ambitious, and limiting to the linguist, who was free to use only those structures which had already been defined by procedures such as that described above, and not anything else from his knowledge of language. He proposed vast methodological changes: Linguistic intuitions are a valid source of scientific evidence; the aim of linguistics was not to define language, but rather to explain formally the regularities which pertained in language which allow us to assign structural descriptions to a potentially infinite set of sentences using a finite computational device (the generative enterprise), and to explain the syntactic relationship between various forms of the 'same' sentence (e.g. passive form, interrogative form, and so on) (the transformational enterprise).

It may appear that the empiricist enterprise described in this thesis has been tried, tested, and found wanting already. This is not the case. The closest method from structural linguistics to this method was the 'frame and focus' method outlined above. This differs from the techniques described here both linguistically and in principle. Linguistically, although frame-and-focus methods are derived from the replacement criterion, they are relatively weak in that the frames cannot be derived in a linguistically naive way, knowledge of what constitutes a valid sentence is assumed (e.g. from an informant), and although they might in principle be able to be used to define such classes, they cannot be thought of as an accurate model of how a child might acquire such structure, nor are they in principle generalisable to the learning of non-linguistic structure.

I shall now review some of the arguments which have been proposed against an empiricist

approach to learning linguistic structure. It is very difficult to prove a negative, so the arguments are of necessity rather vague.

1. No finite corpus can contain every sentence, or even every form of sentence, so corpus based induction is impossible. In *Syntactic Structures* (1957) Chomsky states that the notion of 'grammatical' cannot be identified with 'high order statistical approximation'. As an example he suggests that *green ideas sleep furiously* will have a very low probability for semantic, not grammatical reasons. He goes on to conclude that "I think we are forced to conclude that grammar is autonomous and independent of meaning, and that probabilistic models give no particular insight into some of the basic problems of syntactic structure."

   This is either a straw man, or simply wrong. Just because *green ideas sleep furiously* will never occur in a corpus does not mean that its sentencehood can't be induced *in principle* by the techniques described here, and yet its improbability on semantic grounds also be induced. In other words, there is no reason why the statistical prediction model should not split semantic and pragmatic from grammatical reasons about why a sequence of words is more or less probable. Indeed, if syntax were truly *statistically* independent of semantics, one would expect a good statistical predictor of a language to be modular with respect to such a distinction, and therefore the syntax/semantics distinction would be preserved. Chomsky did not investigate the hypothesis that grammar is the structure that remains when meaning has been controlled for, and seems to have ignored the possibility that sophisticated statistical models can be decomposed into units which correspond to the syntax/semantics modules of traditional linguistic theory.

2. Even if it is possible *in principle* to use bottom-up techniques to infer a grammar, the child does not have available a sufficiently large corpus to use these techniques to infer a grammar.

   This may be so, but it is an empirical statement, and can only really be answered by trying to infer a grammar using such techniques. To have got this far, it seems that about 30,000,000 words are needed. It may well be possible that this technique can be speeded up (for instance by using extra-linguistic information,

such as semantic representations of the world as Pinker suggests), but over a period of a couple of years, it is not at all unlikely that a child will have been exposed to this number of words[1].

Consequently, this is not a good reason to suggest that such bottom-up techniques cannot learn language in principle.

3. Evolution would tend to favour systems which had a strong innate component, since theories with a strong innate component would take far less time to learn. Why attribute language learning to a sophisticated empiricist learning system, rather than millions of years of evolution?

   There are three answers to this point. The first is that evolution and learning are both best understood as search systems, searching for efficient and useful ways to represent and process information about the world. It is true that evolution has had far more time to learn the regularities which exist in nature, but it is not true to say that what it learns is in some sense totally arbitrary, and could not be derived from simple data driven learning systems. For instance, consider object based descriptions of the world. In one way of thinking, objects do not exist to our senses — they are simply a manifestation of an innate (and hence evolutionarily successful) way of viewing the world. However, they do embody many statistical regularities. The orientation and position of a solid object can be defined by 5 parameters. These parameters allow us to make predictions about all the parts of the object. Conversely, every part of an amorphous blob (which constantly changes) needs a separate description, and consequently many more parameters are needed to describe it. Thus an object based description is far more efficient, and exploits the statistical regularities due to the solidity and connectedness of objects, and this promises to be statistically derivable.

   The second answer is that in learning language, systems have not had "millions" of years to develop — there is no evidence of language before homo-sapiens, and no evidence of homo-sapiens before 1 million years ago, or 50,000 generations. Whatever language specific procedures have developed have developed in a very

[1] if a child is exposed to adult conversation at an average of 2 words per second for 5 hours a day, it will hear this number of words in 2.5 years.

short space of time on an evolutionary scale. It seems much more likely that small adaptations to already extant learning processes occurred, rather than the development of highly complex, language specific learning systems: evolution has had a great deal more time to find sophisticated learning strategies than sophisticated language learning strategies, and since a sophisticated learning strategy can fit the data to which the learning agent is exposed more accurately than an innate evolutional system, which has come to be innate by exposure to data which the particular agent which inherited it will never need to process. Indeed, Chomsky himself acknowledges such arguments, saying (Chomsky 1972)

> [an innate language faculty] poses a problem for the biologist, since, if true, it is an example of the true 'emergence' – the appearance of a qualitatively different phenomenon at a specific stage of complexity of organisation.

He then goes on to suggest that any innate language capability is the by-product of other evolutionary forces, such as an increase in overall brain size and constraints on how large nervous systems must be organised. According to Pinker & Bloom (1990), the famous contemporary evolutionary theorist Gould (1982) also takes this position, and believes that the brains of higher animals from monkeys onwards have become adapted to increase the amount of *learning* which goes on in the learnt/innate divide. Strange (although not impossible), then, that if this is the case language should have come about through an increase in the innate component, rather than as an epiphenomenon of a more efficient learning system. Clearly there are selectional advantages for language, so it might have evolved as a separate system. But linguistic abilities are so highly correlated with other cognitive abilities, even taking environment into account, that it seems likely that there is a common genetic explanation for this fact.

Thirdly, we can compare linguistic processing to the processing of other sorts of information. We know that, except in the simplest of organisms, there is a moderately large learning component in visual development (Marr 1982). We also know that there is a large innate component in visual processing, but that if particular areas of the brain are damaged that vision does not develop, so the innateness is tied to certain physical structures of the brain. We also know

that linguistic processing can develop in many areas of the brain if the brain is damaged early in life, making any hypothesis about pre-wired brain structures, such as might be supposed to exist in a highly nativist explanation of language learning, less plausible, and a general learning system, such as might be supposed to exist in many areas of the brain, more plausible.

Fourthly, surely the logical conclusion of an innate position is that there would be just one language, or at least just one syntactic structure, as there is with bees (albeit with different dialects), since this would be the easiest to learn, and need the least data-dependent fitting. As it is, we find that the vocabulary, and more tellingly the structure of all languages change constantly, and are consequently harder to learn.

Clearly, the empiricist position should not be overstated. No evidence has been presented that the subtle structures necessary to represent language can be acquired using the methods presented in this thesis, although it seems likely that much more structure than has so far been elucidated can be recovered. Secondly, no proposals have been made about how generative linguistic regularities can be extracted from the structure which has been uncovered, although work on stochastic context-free grammars (Fujisaki et al. 1989) uncovers just this sort of generative regularity, and so one might suspect that appropriate statistical techniques can be defined which find generative regularities.

Nevertheless, as a viable alternative to the innateness of a specialised language acquisition device, and one which agrees with the views of at least one leading evolutionary theorist, the hypothesis that language might be learned by a generalised learning system deserves a close re-examination.

9.3 Future Work

The suggestion that knowledge of language can be acquired by empirical investigation is by no means new. Indeed, it was the subject of much research in the so-called 'structuralist' paradigm of linguistics. What is new is the availability of fast and powerful computers to process language, and of very large text corpora as data.

Clearly, there are many ways in which future work could expand on the results presented here.

- Different languages could be examined in much the same way that English was for this thesis. Work by Kneser & Ney (1991) is suggestive that self-organising techniques will be able to find structure in German (a scrambling rather than a fixed word order language, where case marking can be used to denote argument role, rather than word order). Other European languages might be analysed in this way, since large natural language corpora either exist, or will soon exist, for all of them. Free word order languages such as Latin might also be analysed in this way, but perhaps in such morphologically complex languages, statistical morphological regularities might be used to cluster words, rather than word order. Thus word order is just one source of statistical regularity for uncovering a categorisation, and this will work best in fixed word order languages where word order determines syntactic structure.

- A finer grained classification of categories could be used to derive more finely grained structure. The choice of level to cut the dendrograms to find a classification is, to a large extent, arbitrary. A finer grained categorisation might be expected to give rise to a finer grained categorisation of sequences, at the expense of needing a larger corpus size to find reliable statistics. For a fixed corpus size, it may be possible to define an optimal categorisation so that the fineness of the grain of the categorisation is balanced with the size of the corpus to produce the finest grained statistically reliable regularities.

- It will be necessary to find *generative* regularities from the finite regularities uncovered by the classification techniques described here. Concentrating on short sequences is statistically advantageous because short sequences are relatively more common in a corpus of a fixed length than long ones (there are more possible 100 word sentences than possible 6 word sentences). It may be possible to define procedures which exploit the structure of short sequences which has been found, to infer rules which generalise to arbitrarily long sentences. Alternatively, it may be possible to use the structure uncovered for short sequences to initialise the para-

meters of a standard generative statistical model, such as a stochastic context free grammar, so that re-estimation procedures for such a generative model converge more quickly to a better solution than a random setting.

- Since nothing in the learning paradigm described in this thesis is entirely specific to natural language, it will be interesting to see whether analogous techniques can be used to derive structure in any other domains.

- There are a number of related techniques in the literature. One very promising one is the idea that in finding procedures which *compress* a source of data, it is possible to analyse these procedures, or the compressed forms, to uncover regularities (Wolff 1991). The classification procedures described in this thesis could be used as a component of such a system. Wolff (1977, 1978) has already shown that a significant amount of orthographic and syntactic structure can be explicated by finding procedures which efficiently compress a corpus, and Ellison (1991, 1992) has shown that phonological information can be explicated in the same way. It is possible that the processes of classification I have used might be employed as a component in a data compression system which might be able to explicate generative rules.

- The techniques described here make use only of distributional evidence, and no other information about the linguistic function of words and phrases, such as might be provided by semantic knowledge as Pinker proposes. Indeed, Powers (1983, 1989) has extensively argued that cross-domain regularities can be exploited to help find structure, and it is just the sort of information about complimentisation patterns as might be provided by knowledge of the meaning of words which will be of greatest use in identifying sentences which are not missing a compliment from sentences which are. Can finer grain linguistic structure be derived by using non-distributional regularities?

I share the view outlined by Powers & Daelemans (1992) that a considerable amount of the structure of natural language might be learnt by unsupervised, self-organising, statistical techniques, but draw the lesson that concentrating on statistically motivated techniques, rather than ad-hoc ones, is the approach most likely to yield success.

Appendix A

Bibliography

Adriaans, P. (1992) Bias in Syntactic Learning. Paper presented at Aberdeen University, Scotland, UK.

Altmann, G. (1980) Prolegomena to Menzerath's Law. *Glottometrica 2* ed. R. Grotjahn, Bochum, 1 – 10.

Altmann, G. & M. H. Schibbe (1989) *Das Menzerathsche Gesetz in Informationsverarbeitenden Systemen* Hildesheim, Zurich, New York.

Anderson, J. R. (1976) *Language, Memory & Thought.* Hillsdale Erlbaum.

Barry, G. & M. Pickering (1989) Dependency and Constituency in Categorial Grammar. *Studies in Categorial Grammar.* G. Barry & G. Morrill (Eds), Centre for Cognitive Science Working Papers, 23–46, University of Edinburgh, Scotland.

Baum, L. E., T. Petrie, G. Soules & N. Weiss (1970) A Maximisation Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *Ann. Math. Stat.* **41** 164–171.

Bechtel, G. G. (1976) *Multidimensional Preference Scaling.* The Hague, Mouton.

Bloomfield, L. (1933) *Language.* Holt, Rinehart and Winston, New York.

Booth, T. (1969) Probabilistic Representation of Formal Languages. In *Tenth Annual IEEE Symposium on Switching and Automata Theory.*

Bresnan, J. (1982) *The Mental Representation of Grammatical Relations.* MIT press.

Brill, E., D. Magerman, M. Marcus & B. Santorini (1990) Deducing Linguistic Structure from the Statistics of Large Corpora. in *DARPA Speech and Natural Language Workshop.* Morgan Kaufmann, Hidden Valley, Pennsylvania.

Brown, P., J. Cocke, S. Della Pietra, V. J. Della Pietra, F. Jelinek, R. L. Mercer & P. Roossin (1988) A Statistical Approach to Language Translation *Proceedings of COLING 88* pp. 71–76.

Brown, P., V. J. Della Pietra, P. V. deSouza, J. C. Lai & R. L. Mercer (1990) *Class Based Language Models.* IBM Technical Report, York Town Heights, NJ.

Clark, R. (1990) Causality and Paramater Setting *Behavioural and Brain Sciences.* June.

Clark, R. & I. Roberts (1991) A Computational Model of Language Learning and Language Change. Technical report in Formal and Computational Linguistics, No. 3. Departmente de Linguistique, Universite de Geneve.

Cleeremans, A., D. Servan-Schrieber & J. L. McClelland (1989) Finite State Automata and Simple Recurrent Networks. *Neural Computation*, 1, 372–381.

Chomsky, N. (1957) *Syntactic Structures.*, The Hague, NL: Mouton.

Chomsky, N. (1965) *Aspects of the Theory of Syntax.* MIT press, Boston, Mass.

Chomsky, N. (1972) *Language and Mind.* Harcourt, Brace and World.

Chomsky, N. (1975) *Reflections on Language.* Pantheon Books, New York.

Chomsky, N. (1980) *Rules and Representations.* Columbia University Press.

Chomsky, N. (1986) *Knowledge of Language: its Nature, Origin and Use.* Praeger, NY.

Church, K. W. (1988) A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. *Proceeding of the Second Conference on Applied Natural Language*, Austin, Texas. 136–143.

Church, K. W. (1992) *Parts of Speech Tagging* Paper presented at the Fifth Annual

CUNY Conference on Human Sentence Processing.

Churchland, P. S. (1978) Fodor on Language Learning. *Synthese* **38** 149–159.

Dowty, D. R., R. E. Wall & S. Peters (1981) *Introduction to Montague Semantics.*, Dordrecht: Reidel.

Durbin, R. & D. J. Willshaw (1987) An Analogue Approach to the Travelling Salesman Problem using an Elastic Net Method. *Nature.* **326** 689–691.

Ellison, T.M. (1991) Discovering Planar Segregations. In D. Powers & L. Reeker (eds) *AAAI Spring Symposium on Machine Learning of Natural Language and Ontology.* 42–47.

Ellison, T.M. (1992) Learning Vowel Harmony. In W. Daelemans & D. Powers (eds) *Background and Experiments in Machine Learning of Natural Language.* ITK, Tilburg, NL. 205–227.

Elman, J. L. (1989) Structured Representations and Connectionist Models. *Proceedings of the Cognitive Science Society of America* 17–23.

Elman, J. L. (1990) Finding Structure in Time. *Cognitive Science*, 14, 179–211.

Finch, S. P. & N. Chater (1991) A Hybrid Approach to the Automatic Learning of Linguistic Categories. *Artificial Intelligence and Simulated Behaviour Quarterly.* **78** 16–24.

Finch, S. P. & N. Chater (1992a) Bootstrapping Syntactic Categories using Statistical Methods. In W. Daelemans & D. Powers (eds) *Background and Experiments in Machine Learning of Natural Language.* ITK, Tilburg, NL. 229–236.

Finch, S. P. & N. Chater (1992b) Bootstrapping Syntactic Categories. *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society of America.* Bloomington, Indiana. 820–825.

Finch, S. P. & N. Chater (1992c) Unsupervised Methods for Finding Linguistic Categories. *Proceedings of the International Joint Conference in Neural Networks.*, Brighton, UK.

Finch, S. P. & N. Chater (forthcoming) Language Learning Using Statistical methods. In M. R. Oaksford & G. D. A. Brown (eds) *Neurodynamics and Psychology*, Academic Press.

Fodor, J. A. (1975) *The Language of Thought.* Thomas Crowell, New York.

Fodor, J. A. (1983) *Modularity of Mind.* MIT press.

Fong, S. & R. Berwick (1992) *Parsing English and Japanese with Principles and Parameters Theory.* Paper presented at the Fifth Annual CUNY Conference on Human Sentence Processing.

Fujisaki, T., F. Jelinek, J. Cocke, E. Black, T. Nishino (1989) Probabilistic Parsing Methods for Sentence Disambiguation. In *Proceedings of the International Parsing Technologies Workshop.* Carnegie-Mellon University, Pittsburgh.

Garside, R. G. Leech & G. Sampson (1987) *The Computational Analysis of English–A Corpus Based Approach* Longman.

Gazdar, G., E. Klein, G. Pullum, I. Sag (1985) *Generalized Phrase Structure Grammar.* London, Blackwell.

Geman, S. & D. Geman (1984) Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE trans. on Computation.* IEEE 0162-8828/84/1100-0721$01.00

Gold, E. M. (1967) Language Identification in the Limit. *Information and Control* **16:** 447–474.

Golden, R. M. (1987) *A Probabilistic Computational Framework for Neural Network Models.* Unpublished Ms.

Gould, S. J. (1987) The Limits of Adaption: Is Natural Language a Spandrel of the Human Brain? Paper presented to Centre for cognitive Science, MIT, October 1987.

Halliday, M. A. K. (1961) Categories of the Theory of Grammar. *Word* **17:3** 241 – 292.

Hammersley, J. M. & D.C. Handscomb (1965) *Monte Carlo Methods* New York, Wiley.

Harris, Z. (1951) *Methods in Structural Linguistics.* Chigago University Press, Chicago.

Hebb, D. O. (1949) *The Organisation of Behaviour: A Neuropsychological Theory.* Wiley, NY.

Hertz, J. A., A. Krogh & R. G. Palmer (1991) *Introduction to the Theory of Neural Computation.* Redwood City, Calif: Addison-Wesley.

Hettmansperger, T. P. (1984) *Statistical Inference Based on Ranks.*, New York: Wiley.

Hinton, G. E., T. J. Sejnowski & D. H. Ackley (1984) *Boltzman Machines: Constraint Satisfaction Networks that Learn.* Technical report CMU-CS-84-119, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

Hopfield, J. J. (1982) Neural Networks and Physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA.* 2554–2558.

Huang, X. D., Y. Ariki & M. A. Jack (1990) *Hidden Markov Models for Speech Recognition.* Edinburgh University Press.

Hughes, J. (1992) *Automatically Acquiring a Classification of Words.* Manuscript.

Hughes, J. & E. S. Atwell (1993) Automatically Acquiring and Evaluating a Classification of Words. IEE digest 1993/092: *Garmmatical Inference: Theory, Applications and Alternatives.*

Jelinek, F. (1986) *Self Organized Language Modelling for Speech Recognition.* IBM Technical report, T. J. Watson Research Centre, Yorktown Heights, N.Y.

Jelinek, F., J. D. Lafferty & R. L. Mercer (1990) Basic Methods of Probabilistic Context Free Grammars. Technical Report RC 16374 (72684), IBM, Yorktown Heights, New York.

Katz, J. J. & J. A. Fodor (1963) Structure of a Semantic Theory. *Language*, 39, 170–210.

Kemler Nelson, D. K. Hirsh-Parak, P. Jusczyk & K. Cassidy (1989) How Prosodic Cues in Motherese might Assist Language Learning. *Journal of Child Language.* **16** 55–68.

Kiss, G. R. (1972) Grammatical Word Classes: A Learning process and its Simulation. *Psychology of Learning and Motivation* **7** 1–41.

Kneser, R. & H. Ney (1991) Forming Word Classed by Statistical Clustering for Statistical Language Modelling. in Proceedings of QUALICO 1, Trier, Germany.

Kohonen, T. (1982) Self Organised Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, **43**, 59-69.

Kolmogorov, A.N. (1965) Three Approaches to the Quantitative Definition of Information. *Problems in Information Transmission.* **1**, 4–7.

Kral, K. & I. A. Meinertzhagen (1989) Anatomical Plasticity Synapses in the Lamina of the Optic Lobe of the Fly. *Phil. Trans. of the Royal Society.* **323** 155–183, London.

Kuhn, T. (1970a) *The Structure of Scientific Revolutions, 2nd Edition* University of Chicago Press, Chicago.

Kuhn, T. (1970b) Logic of Discovery or Psychology of Research? In I. Lakatos & A. Musgrave (Eds) *Criticism and the Growth of Knowledge.* 1–23. University of Chicago Press, Chicago.

Kupiec, J. (1992) Robust Part-of-Speech Tagging using a Hidden Markov Model. *Computer Speech & Language.* **6** 3:225–242.

Lachter, J. & T. G. Bever (1988) The Relation between Linguistic Structure and Associative Theories of Language Learning: A Constructive Critique of some Connectionist Learning Models. *Cognition*, 28, 73–193.

Lakatos, I. (1970) Falsification and the Methodology of Research Programmes. In I. Lakatos & A. Musgrove (eds) *Criticism and the Growth of Knowledge.* 91–196, University of Chicago Press, Chicago.

Lambek, J. (1958) The Mathematics of Sentence Structure. *American Mathematical Monthly.* **65** 154 – 170.

Lehmann, E. L. (1975) *Nonparametrics: Statistical Methods Based on Ranks.*, San Francisco: Holden-Day.

Liberman, M., D. E. Walker, S. Warwick & A. Zampolli (1991) *ACL/DCI CD-ROM 1.* University of Pennsylvania, PA.

Locke, J. (1690) *An Essay Concerning Human Understanding.*

McCulloch, W. S. & W. Pitts (1943) A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics* **5** 115–133.

McDermott, D. V. & J. Doyle (1980) Non-monotonic Logic I. *Artificial Intelligence* **13** (no. 1,2); 41–72.

Marcus, M. (1991) Distituency Grammar. *Proceedings of the 1991 AAAI symposium on Statistical Methods for Natural Language.*

Markov, A. A. (1913) An Example of Statistical Investigation in the Text of Eugen Onyegin Illustrating Coupling of Tests in Chains. *Proceedings of the Academy of Science of St Petersburgh VI* Series, 7, 153-162.

Marr, D. (1982) *A Computational Investigation in the Human Representation of Visual Information.* Freeman, San Francisco.

Maratsos, M. (1982) The Child's Reconstruction of Grammatical Categories. In E. Wanner & P. Mussen (eds) *Language Acquisition.* Cambridge University Press, NY.

Marsatos, M. (1990) Are Actions to Verbs as Objects are to Nouns? On the Differential Semantic Bases of Form, Class, Category. *Linguistics* **28:** 1351–1379.

Mandelbrot, B. (1953) An Informational Theory of the Statistical Sructure of Language. In W. Jackson (ed.) *Communication Theory.* London, Butterworths, 486–502.

Mendeleev, D. I. (1891) *Grundlagen der Chemie.* St. Petersburg C. Ricker 1891

Mercer, R. (1992) *With Friends Like Statistics, Who Needs Linguistics?* Talk presented at the Fifth Annual CUNY Conference on Human Sentence Processing.

Minsky, M. & S. Papert (1969) *Perceptrons* MIT press, Cambridge, Mass.

Morgan, J. & E. Newport (1981) The Role of Constituent Structure in the Induction of an Artificial Language. *Journal of Verbal Learning and Verbal Behaviour.* **20:** 67–85.

Ortony, A. (1979) Beyond Literal Similarity. *Psychological Review* **86** 161–180.

Osherson, D. & E. E. Smith (1981) On the Adequacy of Prototype Theory as a Theory

of Concepts. *Cognition* **9** 35–58.

Osherson, D., M. Stob & S. Weinstein (1986) *Systems That Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists.* Cambridge, Mass: MIT Press.

Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo: Morgan Kaufmann.

Pereira, F. C. N. & Y. Schabes (1992) Inside-Outside Reestimation from Partially Bracketed Corpora *Fifth DARPA Speech and Natural Language Workshop, February, 1992*

Pinker, S. (1979) Formal Models of Language Learning. *Cognition* **7**.

Pinker, S. (1984) *Language Learnability and Language Development.* Cambridge, Mass: Harvard University Press.

Pinker, S. (1987) The Bootstrapping Problem in Language Acquisition. In B. MacWhinney (ed.) *Mechanisms of language Acquisition.* Hillsdale: Erlbaum.

Pinker, S. & P. Bloom (1990) Natural Language and Natural Selection. *Behavioural and Brain Sciences* **13** 707–734.

Pinker, S. & A. Prince (1988) On Language and Connectionism: Analysis of a Parallel Distributed Processing Model of Language Acquisition, *Cognition*, 28, 195–247.

Popper, K. R. (1959) *The Logic of Scientific Discovery.* London, Hutchinson.

Popper, K. R. (1970) Normal Science and its Dangers. In I. Lakatos & A. Musgrave (Eds) *Criticism and the Growth of Knowledge.* 51–58. University of Chicago Press, Chicago.

Powers, D.M.W. (1983) Neurolinguistics and Psycholinguistics as a Basis for the Computer Acquisition of Natural Language. *SIGART newsletter* 29–34.

Powers, D.M.W. & C. Turk (1989) *Machine Learning of Natural Language.* Research Monograph, Springer-Verlag.

Powers, D.M.W & W. Daelemans (1992) SHOE: The Extraction of Hierarchical Structure for Machine Learning of Natural Language. Project Summary In W. Daelemans

& D. Powers (eds) *Background and Experiments in Machine Learning of Natural Language.* ITK, Tilburg, NL. 125–159.

Powers, D.M.W. (1992) On the Significance of Closed Classes and Boundary Conditions: Experiments in Lexical and Syntactic Learning. In W. Daelemans & D. Powers (eds) *Background and Experiments in Machine Learning of Natural Language.* ITK, Tilburg, NL. 245–266.

Pullum, G. K. & G. Gazdar (1982) Natural Languages and Context Free Languages.

Radford, A. (1988) *Transformational Grammar.* 2nd Edition, Cambridge: Cambridge University Press.

Redington, F. M. (1992) *A Statistical Approach to Syntax Acquisition.* Masters Thesis, University of Edinburgh, Dept. of AI.

Rissanen, J. (1978) Modelling by Shortest Data Description. *Automatica.* **14**, 445–71.

Rosch, E. (1973) Natural Categories. *Cognitive Psychology* 4 328–350.

Rosch, E. (1975) Family Resemblance: Studies in the Internal Structure of Categories. *Cognitive Psychology* **7** 573 – 605.

Rosenblatt, F. (1959) Two Theorems of Statistical Separability in the Perceptron. in *Proceedings of the National Physical Laboratory* HM Stationary Office, London, 421–456.

Rumelhart, D. E., G. E. Hinton & R. J. Williams (1986) Learning Representations by Back-propagating Errors. *Nature* **323:** 533–536.

Sampson, G. (1992) Statistical Linguistics. Article in *The Oxford International Encyclopedia of Linguistics.* Oxford University Press, New York.

Sampson, G. (1992) *The* SUSANNE *Corpus of American English. Release 1.* School of Cognitive & Computing Sciences, University of Sussex, Brighton, UK.

Schabes, Y. (1992) Stochastic Tree adjoining Grammars. in *Proceedings of the Fifth DARPA Speech and Natural Language Workshop* Feb. 1992.

Scholtes, J. C. (1991a) Kohonen's Self Organising Map Applied Towards Natural Language Processing. *Proceedings of the CUNY 1991 Conference on Human Sentence Pro-*

cessing, Rochester.

Scholtes, J. C. (1991b) Using Extended Feature Maps in a Language Acquisition Model. *Proceedings of the 2nd Austrailian Conference on Neural Networks*

Scott, D. (1981) *Lectures on a Mathematical Theory of Computation.* Research Monograph PRG-19, Oxford University Computing Laboratory, Oxford.

Shannon, C. E. (1948) A Mathematical Theory of Communications. Bell Systems Technical Journal, **27**, pp.379-423.

Shannon, C. E. & W. Weaver (1949) *The Mathematical Theory of Communication.* United Independent Press.

Shannon, C. E. (1951) Prediction and Entropy of Printed English. Bell Systems Technical Journal, **30**, pp. 50-64.

Shieber, S. M. (1984) Evidence against the Context-Freeness of Natural Language. *Linguistics and Philosophy.*

Smith, E. E. & D. L. Medin (1981) *Categories and Concepts.* Harvard University Press, Cambridge, Mass.

Solomonoff, R. (1964) A Formal Theory of Inductive Inference. *Information and Control.* **7**, 1–22 & 224–254

Sokal, R. R. & P. H. A. Sneath (1963) *Principles of Numerical Taxonomy.* San Francisco: W.H. Freeman.

Spackman, K. A. (1991) Maximum Likelihood Training of Connectionist Models: Comparison with Least Squares Back-Propagation and Logistic Regression. *Proceedings of the 15th Symposium on Computer Applications in Medical Care*, p. 285.

Steedman, M. (1985) Combinators and Grammars. *Proceedings of the Tucson Conference on Categorial Grammar.* Tucson.

Stoy, J. E. (1977) *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory*, MIT press, Mass.

Svartvik, J. & R. Quirk (1980) *A Corpus of English Conversation.* Lund: LiberLaro-

medel Lund.

Tversky, A. & I. Gati (1978) Studies of Similarity. in *Cognition and Categorization.* Hillsdale, NJ., 79–98.

Valiant, L. G. (1984) A Theory of the Learnable. *Communications of the ACM.* **27.**

Viterbi, A. J. (1967) Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. In *IEEE Transactions on Information Theory.* 260–269.

Wallace, C.S. & D. M. Boulton (1968) An Information Measure for Classification. *Computer Journal.* **11** 185–195.

Wexler, K. & P. Cullicover (1980) *Formal Principles of Language Acquisition.* MIT Press.

Wharton, R. (1974) Approximate Language Identification. *Information and Control.* **26** 236–255.

Willshaw, D. J. & C. von der Malsburg (1976) How Patterned Neural Connections can be Set Up by Self-Organization, *Proceeding of the Royal Society.* **194** pp. 431–445.

Wittgenstein, L. (1953) *Philosophical Investigations.* Blackwell, Oxford.

Wolff, J. G. (1977) The Discovery of Segments in Natural language, *The British Journal of Psychology.* **68** 97–106.

Wolff, J.G. (1978) Recoding of Natural Language for Economy of Transmission *The Computer Journal.* **21:1** 42–44.

Wolff, J.G. (1991) *Towards a Theory of Cognition and Computation.* Chichester: Ellis Horwood.

Zipf, G. K. (1935) *The Psycho-biology of Language.* Boston: Houghton Mifflin.

Zipf, G. K. (1949) *Human Behaviour and the Principle of Least Effort.* Cambridge, Mass.

Appendix B

A Simple CFG Used in Classification

This appendix lists the stochastic context free grammar used to derive the classification of chapter 6. To interpret it, the numbers refer to weights, and to derive a probability from a weight, sum the weights of all the rules assocoated with a non-terminal symbol on the left hand side, and divide the weight of a particular rule by this sum. This is the probability that an instance of the non-terminal symbol will be expanded by that particular rule.

Rules which introduce terminal symbols have an asterisk after the arrow.

```
s -- 1.0 --> nps vps
s -- 1.0 --> npp vpp

nps -- 1.0 --> dets nbars
npp -- 1.0 --> detp nbarp
npp -- 0.3 --> nbarp
nps -- 0.3 --> pn

nbars -- 1.5 --> ap nouns
nbars -- 1.0 --> ap nouns pp
```

```
nbars -- 1.0 --> nouns pp

nbarp -- 1.5 --> ap nounp
nbarp -- 1.0 --> ap nounp pp
nbarp -- 1.0 --> nounp pp

nbars -- 1.0 --> nouns rel vps
nbarp -- 1.0 --> nounp rel vpp

nbars -- 3.0 --> nouns
nbarp -- 3.0 --> nounp

np -- 1.0 --> nps
np -- 1.0 --> npp
vp -- 1.0 --> vps
vp -- 1.0 --> vpp

vbars -- 1.0 --> vts np
vbars -- 1.0 --> vis
vbars -- 1.0 --> vds np np

vbarp -- 1.0 --> vtp np
vbarp -- 1.0 --> vip
vbarp -- 1.0 --> vdp np np

vps -- 2.0 --> aux vbarp
vps -- 1.0 --> aux adverb vbarp
vps -- 1.0 --> adverb aux vbarp
vps -- 6.0 --> vbars
vps -- 1.0 --> adverb vbars
```

```
vpp -- 1.0 --> aux vbarp
vpp -- 1.0 --> aux adverb vbarp
vpp -- 1.0 --> adverb aux vbarp
vpp -- 4.0 --> vbarp
vpp -- 1.0 --> adverb vbarp


pp -- 1.0 --> prep np

ap -- 1.0 --> adverb ap
ap -- 3.0 --> adjective


dets -- 3.0 --> * the
dets -- 1.0 --> * a
dets -- 1.0 --> * her
dets -- 1.0 --> * my

detp -- 3.0 --> * the
detp -- 1.0 --> * her
detp -- 1.0 --> * all
detp -- 1.0 --> * some

pn -- 1.0 --> * edinburgh
pn -- 2.0 --> * fred
pn -- 3.0 --> * steve

adverb -- 2.0 --> * carefully
adverb -- 1.0 --> * happily
adverb -- 2.0 --> * deliberately
adverb -- 1.5 --> * quickly
```

```
adverb -- 1.9 --> * slowly
adverb -- 2.5 --> * probably
adverb -- 1.1 --> * totally
adverb -- 3.0 --> * really

adjective -- 2.0 --> * big
adjective -- 1.4 --> * green
adjective -- 0.5 --> * real
adjective -- 1.2 --> * polite
adjective -- 1.7 --> * happy

nouns -- 1.0 --> * bag
nouns -- 1.0 --> * book
nouns -- 1.0 --> * clock
nouns -- 1.0 --> * cat
nouns -- 1.0 --> * dog
nouns -- 1.0 --> * computer
nouns -- 1.0 --> * hamster
nouns -- 1.0 --> * man
nouns -- 1.0 --> * hat
nouns -- 1.0 --> * rabbit

nounp -- 1.0 --> * books
nounp -- 1.0 --> * dogs
nounp -- 1.0 --> * men
nounp -- 1.0 --> * cats
nounp -- 1.0 --> * clocks
nounp -- 1.0 --> * people
nounp -- 1.0 --> * women
nounp -- 1.0 --> * computers
nounp -- 1.0 --> * hamsters
```

```
nounp -- 1.0 --> * rabbits

prep -- 1.0 --> * against
prep -- 1.0 --> * of
prep -- 1.0 --> * in
prep -- 1.0 --> * over
prep -- 1.0 --> * to
prep -- 1.0 --> * under
prep -- 1.0 --> * through
prep -- 1.0 --> * from
prep -- 1.0 --> * with
prep -- 1.0 --> * on

rel -- 1.0 --> * that
rel -- 1.0 --> * which

vds -- 1.0 --> * gives
vds -- 1.0 --> * makes
vds -- 1.0 --> * tells
vds -- 1.0 --> * sends
vds -- 1.0 --> * offers

vdp -- 1.0 --> * give
vdp -- 1.0 --> * make
vdp -- 1.0 --> * send
vdp -- 1.0 --> * tell
vdp -- 1.0 --> * offer
vdp -- 1.0 --> * pay

vis -- 1.0 --> * dies
vis -- 1.0 --> * vanishes
```

```
vis -- 1.0 --> * hopes
vis -- 1.0 --> * jumps
vis -- 1.0 --> * runs
vis -- 1.0 --> * sings
vis -- 1.0 --> * walks
vis -- 1.0 --> * miaows
vis -- 1.0 --> * eats
vis -- 1.0 --> * sleeps

vip -- 1.0 --> * die
vip -- 1.0 --> * eat
vip -- 1.0 --> * miaow
vip -- 1.0 --> * sing
vip -- 1.0 --> * sleep
vip -- 1.0 --> * vanish
vip -- 1.0 --> * walk
vip -- 1.0 --> * hope
vip -- 1.0 --> * jump
vip -- 1.0 --> * run

vtp -- 1.0 --> * compute
vtp -- 1.0 --> * need
vtp -- 1.0 --> * throw
vtp -- 1.0 --> * reject
vtp -- 1.0 --> * support
vtp -- 1.0 --> * hit
vtp -- 1.0 --> * like
vtp -- 1.0 --> * want
vtp -- 1.0 --> * kick
vtp -- 1.0 --> * wear
```

```
vts -- 1.0 --> * computes
vts -- 1.0 --> * supports
vts -- 1.0 --> * wants
vts -- 1.0 --> * kicks
vts -- 1.0 --> * likes
vts -- 1.0 --> * throws
vts -- 1.0 --> * hits
vts -- 1.0 --> * rejects
vts -- 1.0 --> * wears
vts -- 1.0 --> * needs


aux -- 1.0 --> * could
aux -- 1.0 --> * will
aux -- 1.0 --> * would
aux -- 1.0 --> * should
aux -- 1.0 --> * might
```

Appendix C

Word Classes

Here is a list of the word classes derived from the analysis of natural language described in chapter 6 by cutting the dendrogram when about 400 remained. These were ordered by frequency, and the most frequent 100 were chosen. The following list, from the most frequent to the least, resulted.

PERIOD COMMA PERIOD

START START

C1 the my your their his our its a an any some several another every these those such each no many most certain

C2 of in on at for with from by into through against about between without under within during via upon towards toward across among beyond regarding

C3 time room way case thing reason moment chance problem situation idea argument statement article book story movie song album film paper letter system program machine game code version product group newsgroup country area language team company line box board car card character body package site field server driver compiler library printer file disk window directory screen command device key table script party band station mode path number level size side job position relationship behavior price solution method tool option word term style ball plane

tank engine unit project event class service school house office city club theory policy organization concept function process format interface environment structure model software data text memory network application user shell terminal modem cable resource database keyboard font utility image header entry routine manager editor magazine controller scheme source reference title description definition distribution connection value purpose status location section string signal sequence pattern page volume frame error item analysis implementation bbs chip bus host client road door wall street tree bar phone tape gun head camera mouse menu clock stack picture piece hole world net war government law church action life culture history race society industry population community army nation universe planet department region question issue discussion thread topic subject defense goal interpretation role flight operation movement difference example opinion response reaction choice decision suggestion attitude conclusion explanation ability effort speech ground court market day night year week month series season morning hour episode scene stage period town summer battle length distance range frequency person woman guy man player girl poster friend mother father wife son child dog family doctor dealer cat brother boy lady baby author owner manual documentation university student bank center ii conference league basis context direction possibility soul union crime information info advice knowledge protection stuff music material equipment money food technology logic education power space energy weight water air oil gas fuel light ram traffic noise research development business security speed quality performance background future production output input bug pointer variable domain password license

C4 is was are were isn't wasn't aren't weren't am

C5 i they we he she you

C6 i've you've we've they've i'd i'll you'll you'd we'll

C7 to

C8 it this me them him her us

C9 would should can could will may might must don't didn't doesn't won't wouldn't can't cannot couldn't shouldn't dont shall does did

C10 get take make give send keep bring put write buy leave sell carry hold pay eat lose add build pick throw pull follow use play read run shoot kill save catch meet reach join draw handle watch allow provide require include create produce receive apply serve wear become spend share fill call turn move change break pass check drop return win fight ride visit count fit avoid remove replace defend prevent protect convince compare connect convert

C11 that what how why

C12 know think believe say feel assume realize mean imagine see find hear remember understand tell ask explain consider accept learn forget teach expect appreciate enjoy discuss ignore recognize choose suggest recommend prove admit deny decide determine define describe mention notice recall

C13 and but or than

C14 have want need do let

C15 as if when where whether because since though until unless before after while

C16 things ones problems questions ideas comments opinions suggestions thoughts details references examples facts errors answers responses replies views feelings experiences systems machines games files programs products books groups words lines characters names numbers cards sources arguments results tools values resources fonts applications functions classes parts images disks tapes issues areas events items versions types options articles messages postings stories papers statements letters points features effects rights rules laws standards weapons actions conditions services methods operations records objects songs movies countries companies sites languages schools teams computers cars commands newsgroups terms elements prices notes instructions discussions differences addresses times cases places reasons ways states nations votes

C17 used made done taken created passed lost changed written built developed called sent brought held sold kept turned moved pulled stopped carried tried started looked worked played picked placed killed caught saved dropped spent followed covered given shown considered allowed paid gone stuck checked included added installed accepted supported removed returned heard seen found noticed discovered told asked learned bought received met missed posted mentioned released discussed reported stated described listed published

C18 good great nice bad hard strong different similar little new local real big small large major special specific normal regular high low full short huge hot heavy minor quick cheap common simple natural legal practical logical moral serious positive negative significant decent solid random critical standard basic objective official dark wild wide deep empty pure false

C19 be been gotten

C20 not only even

C21 out up down off away back over around along together

C22 just simply merely also still probably certainly definitely usually really actually never always generally hardly

C23 old older black white red poor young jewish christian female gay male blue green english french chinese japanese british german russian human personal political physical social religious sexual national technical scientific private foreign western european modern civil medical historical economic cultural internal external electronic serial mental federal musical commercial scsi nuclear virtual digital upper financial electrical educational emotional existing remote american indian arab greek turkish soviet iraqi public military international drug original current recent previous actual initial second third single final main primary central native ancient mathematical traditional formal holy grand electric laser telephone global dynamic chemical

C24 has had got took gets takes makes gave gives saw wrote showed keeps brings needs wants uses

C25 point place name list address copy account state record order form view request date drive port block ship monitor ring rule patch filter document link map test design display base rate approach

C26 go come look live stay sit walk fly stand jump fall die grow try start stop continue begin agree disagree listen speak respond argue happen appear exist occur wait

C27 more less better much enough

C28 computer pc mac amiga sun unix c dos windows news video cd tv radio color hardware postscript graphics internet usenet macintosh magic audio math college star individual animal metal ice binary floppy ethernet cpu os insurance general sales health safety

C29 people women men americans others jews users players students fans parents readers guys folks children kids friends members

C30 available possible necessary interesting important useful difficult dangerous easy fun reasonable appropriate relevant valid fair true obvious apparent illegal acceptable free perfect safe popular effective active successful complex powerful expensive accurate reliable

C31 all both

C32 one

C33 who which whom

C34 very pretty fairly extremely relatively quite too reasonably

C35 using doing getting taking making giving putting sending having buying paying seeing finding watching calling keeping selling changing turning

C36 there

C37 same particular whole entire favorite latest own usual

C38 like

C39 it's i'm you're they're we're he's she's that's there's

C40 anyone anybody someone somebody everyone everybody nobody

C41 other various extra additional multiple people's

C42 message home

C43 something anything nothing everything

C44 going coming looking talking waiting sitting standing listening dealing

C45 few couple lot bit

C46 said thought felt decided claimed realized believed figured suggested assumed meant

C47 came went comes goes starts fell ran turns works runs looks sounds

C48 first last next

C49 work sound end act

C50 post report show answer fix quote match sign

C51 so

C52 right wrong fine ok

C53 seems appears feels says thinks knows knew tells asks

C54 two three four five six ten seven

C55 sure guess bet suppose suspect doubt

C56 well far

C57 fact course least

C58 working running playing moving reading writing driving flying walking

C59 experience sense advantage effect attention interest concern

C60 evidence proof truth meaning reality nature existence belief success loss religion faith

C61 long fast early late cold

C62 interested concerned aware involved responsible familiar

C63 years months weeks days hours minutes

C64 help care

C65 here today

C66 science engineering management physics art computing programming training processing communications communication

C67 israel iraq india america china japan kuwait europe canada taiwan

C68 set cut hit beat shut

C69 now

C70 myself yourself themselves itself himself

C71 able willing ready

C72 x ftp uucp nfs

C73 stupid silly funny strange weird odd nasty excellent wonderful beautiful neat sad amazing

C74 being

C75 ever already

C76 then

C77 hands eyes feet arms eye heart brain blood hair

C78 kind sort

C79 top front middle bottom

C80 completely totally fully entirely somewhat slightly perfectly highly

C81 talk deal worry

C82 best worst

C83 mail email

C84 seem seemed

C85 earth empire gulf media bible usa uk

C86 death sex abortion freedom peace

C87 vote comment reply step

C88 part

C89 saying thinking feeling

C90 support respect trust

C91 above following

C92 apple hp ibm st v

C93 god jesus saddam bush

C94 hope wish

C95 control access

C96 again twice

C97 else

C98 open close clean

C99 self e

C100 near outside behind inside